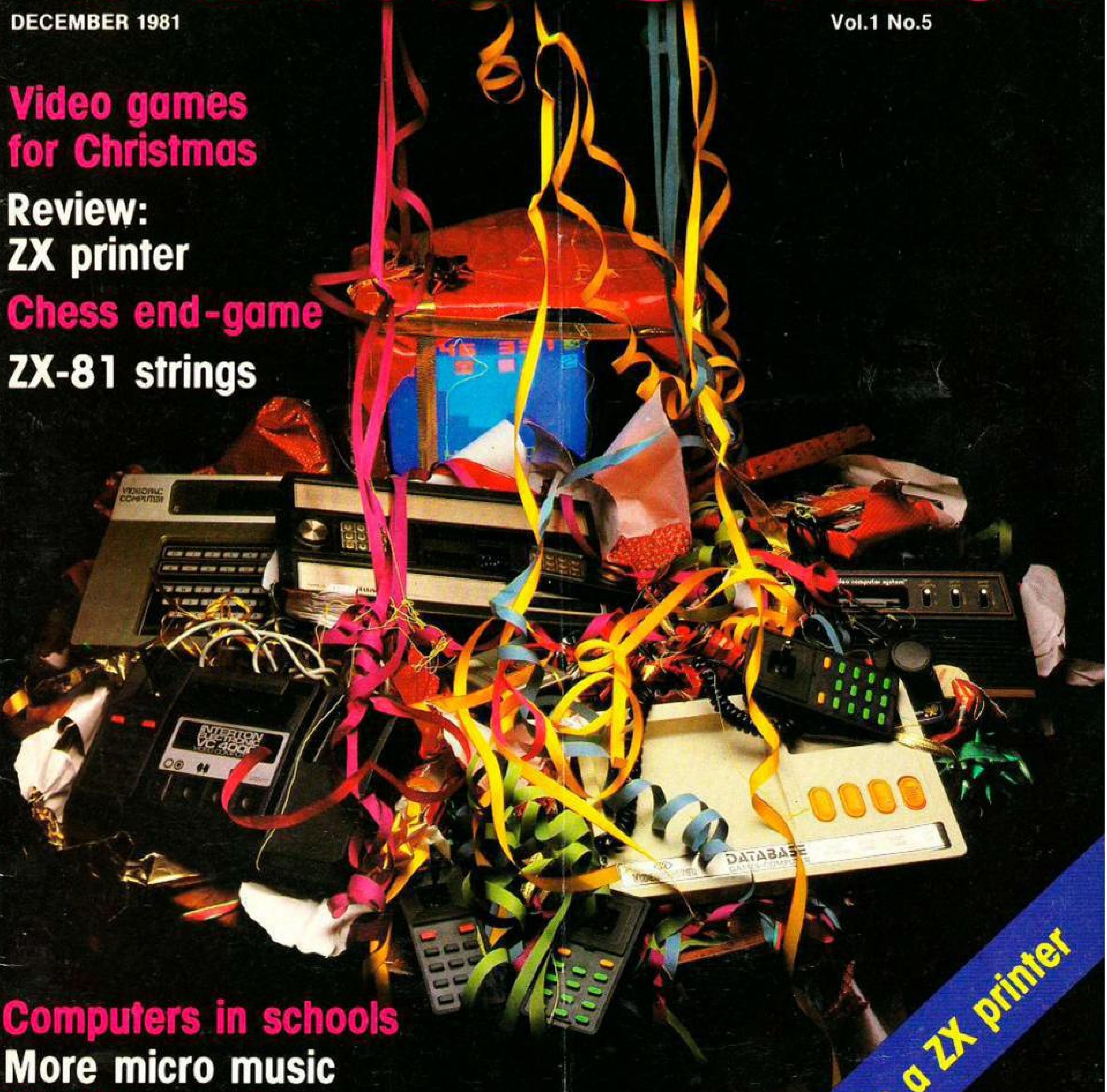
50p

YOUR COMPUTER

DECEMBER 1981

Vol.1 No.5



Video games for Christmas

Review: ZX printer

Chess end-game

ZX-81 strings

Computers in schools

More micro music

Vic-20 cassettes

Win a ZX printer

Make the most of your Sinclair ZX Computer...

Sinclair ZX software on cassette.

£3.95 per cassette.



The unprecedented popularity of the ZX Series of Sinclair Personal Computers has generated a large volume of programs written by users.

Sinclair has undertaken to publish the most elegant of these on pre-recorded cassettes. Each program is carefully vetted for interest and quality, and then grouped with other programs to form a single-subject cassette.

Each cassette costs £3.95 (including VAT and p&p) and comes complete with full instructions.

Although primarily designed for the Sinclair ZX81, many of the cassettes are suitable for running on a Sinclair ZX80 – if fitted with a replacement 8K BASIC ROM.

Some of the more elaborate programs can be run only on a Sinclair ZX Personal Computer augmented by a 16K-byte add-on RAM pack.

This RAM pack and the replacement ROM are described below. And the description of each cassette makes it clear what hardware is required.

8K BASIC ROM

The 8K BASIC ROM used in the ZX81 is available to ZX80 owners as a drop-in replacement chip. With the exception of animated graphics, all the advanced features of the ZX81 are now available on a ZX80 – including the ability to run much of the Sinclair ZX Software.

The ROM chip comes with a new keyboard template, which can be overlaid on the existing keyboard in minutes, and a new operating manual.

16K-BYTE RAM pack

The 16K-byte RAM pack provides 16-times more memory in one complete module. Compatible with the ZX81 *and* the ZX80, it can be used for program storage or as a database.

The RAM pack simply plugs into the existing expansion port on the rear of a Sinclair ZX Personal Computer.



Cassette 1 – Games

For ZX81 (and ZX80 with 8K BASIC ROM)

ORBIT – your space craft's mission is to pick up a very valuable cargo that's in orbit around a star.

SNIPER – you're surrounded by 40 of the enemy. How quickly can you spot and shoot them when they appear?

METEORS – your starship is cruising through space when you meet a meteor storm. How long can you dodge the deadly danger?

LIFE – J. H. Conway's 'Game of Life' has achieved tremendous popularity in the computing world. Study the life, death and evolution patterns of cells.

WOLFPACK – your naval destroyer is on a submarine hunt. The depth charges are armed, but must be fired with precision.

GOLF – what's your handicap? It's a tricky course but you control the strength of your shots.

Cassette 2 – Junior Education: 7-11-year-olds

For ZX81 with 16K RAM pack

CRASH – simple addition – with the added attraction of a car crash if you get it wrong.

MULTIPLY – long multiplication with five levels of difficulty. If the answer's wrong – the solution is explained.

TRAIN – multiplication tests against the computer. The winner's train reaches the station first.

FRACTIONS – fractions explained at three levels of difficulty. A ten-question test completes the program.

ADDSUB – addition and subtraction with three levels of difficulty. Again, wrong answers are followed by an explanation.

DIVISION – with five levels of difficulty. Mistakes are explained graphically, and a running score is displayed.

SPELLING – up to 500 words over five levels of difficulty. You can even change the words yourself.

Cassette 3 – Business and Household

For ZX81 (and ZX80 with 8K BASIC ROM) with 16K RAM pack

TELEPHONE – set up your own computerised telephone directory and address book. Changes, additions and deletions of up to 50 entries are easy.

NOTE PAD – a powerful, easy-to-run system for storing and retrieving everyday information. Use it as a diary, a catalogue, a reminder system, or a directory.

BANK ACCOUNT – a sophisticated financial recording system with comprehensive documentation. Use it at home to keep track of 'where the money goes,' and at work for expenses, departmental budgets, etc.

Cassette 4 – Games

For ZX81 (and ZX80 with 8K BASIC ROM) and 16K RAM pack

LUNAR LANDING – bring the lunar module down from orbit to a soft landing. You control attitude and orbital direction – but watch the fuel gauge! The screen displays your flight status – digitally and graphically.

TWENTYONE – a dice version of Blackjack.

COMBAT – you're on a suicide space mission. You have only 12 missiles but the aliens have unlimited strength. Can you take 12 of them with you?

SUBSTRIKE – on patrol, your frigate detects a pack of 10 enemy subs. Can you depth-charge them before they torpedo you?

CODEBREAKER – the computer thinks of a 4-digit number which you have to guess in up to 10 tries. The logical approach is best!

MAYDAY – in answer to a distress call, you've narrowed down the search area to 343 cubic kilometers of deep space. Can you find the astronaut before his life-support system fails in 10 hours time?

Cassette 5 – Junior Education: 9-11-year-olds

For ZX81 (and ZX80 with 8K BASIC ROM)

MATHS – tests arithmetic with three levels of difficulty, and gives your score out of 10.

BALANCE – tests understanding of levers/fulcrum theory with a series of graphic examples.

VOLUMES – 'yes' or 'no' answers from the computer to a series of cube volume calculations.

AVERAGES – what's the average height of your class? The average shoe size of your family? The average pocket money of your friends? The computer plots a bar chart, and distinguishes MEAN from MEDIAN.

BASES – convert from decimal (base 10) to other bases of your choice in the range 2 to 9.

TEMP – Volumes, temperatures – and their combinations.

How to order

Simply use the order form below, and either enclose a cheque or give us the number of your Access, Barclaycard or Trustcard account. Please allow 28 days for delivery. 14-day money-back option.

sinclair ZX SOFTWARE

Sinclair Research Ltd, 6 Kings Parade, Cambridge, Cambs., CB2 1SN. Tel: 0276 66104.

To: Sinclair Research, FREEPOST 7, Cambridge, CB2 1YY *Please print*

Please send me the items I have indicated below.

Qty	Code	Item	Item price	Total
	21	Cassette 1 – Games	£3.95	
	22	Cassette 2 – Junior Education	£3.95	
	23	Cassette 3 – Business and Household	£3.95	
	24	Cassette 4 – Games	£3.95	
	25	Cassette 5 – Junior Education	£3.95	
	17	*8K BASIC ROM for ZX80	£19.95	
	18	*16K RAM pack for ZX81 and ZX80	£49.95	
		*Post and packing (if applicable)	£2.95	
			Total £	

*Please add £2.95 to total order value **only** if ordering ROM and/or RAM.

I enclose a cheque/PO to Sinclair Research Ltd for £______

Please charge my Access*/Barclaycard/Trustcard no.

**Please delete as applicable.*

Name: Mr/Mrs/Miss

Address:

YOC12

YOUR COMPUTER

Cover photograph by Stephen Oliver. Our thanks to the Silica Shop for lending us the video games for the photograph and the review.

Editor
DUNCAN SCOT

Staff Writer
BILL BENNETT

Production Editor
TOBY WOLPE

Production Assistant
JOHN LIEBMANN

Editorial Secretary
LYNN COWLING

Editorial: 01-661 3500

Advertisement Manager
DAVID LAKE 01-661 3021

Assistant Advertisement Manager
PHILIP KIRBY 01-661 3127

Advertisement Executives
KEN WALFORD 01-661 3139
BILL ARDLEY 01-661 3127

Midlands Office
DAVID HARVETT 021-356 4838

Northern Office
RON SOUTHALL 061-872 8861

Advertisement Secretary
MANDY MORLEY

Publishing Director
CHRIS HIPWELL

Your Computer, Quadrant House, The Quadrant, Sutton, Surrey SM2 5AS.
Typesetting by In-Step Ltd, London EC1.
Printed by Riverside Press Ltd, Whitstable, Kent.

Subscriptions: U.K. £6 for 12 issues.

Published by IPC Electrical-Electronic Press Ltd, Quadrant House, The Quadrant, Sutton, Surrey SM2 5AS. Tel: 01-661 3500 Telex/grams: 892084 BIPRESG.

EDITORIAL

THERE IS A PLAN to impose an extra charge on the cost of blank cassette tapes — the medium most of use to save our lovingly-created programs. One might argue that cassettes are expensive enough already, especially if one always plays it safe and keeps a back-up copy of every program.

The proposal originates from the British Phonographic Institute (BPI), the industry association of record manufacturers. The industry has been in a steady decline since its days of heady success in the sixties. Sales of records, LPs and singles, have fallen and a number of record-producing companies have gone to the wall. The reason, so the BPI argues, is that too many potential record buyers have chosen to break the law of copyright and tape their favourite records. This, the BPI says, robs the industry of its just rewards, and deprives the original recording artists of royalties.

The BPI believes it is powerful and influential enough to persuade the Government to include the levy proposal in the forthcoming revision of the law of copyright.

There are a number of objections to this proposal. The first is that it will cost money to implement and supervise the levy — money which is effectively wasted. Secondly, it will be left entirely to the discretion of the record producers how to distribute the proceeds — there is no guarantee that it will be fair and equitable. Thirdly, some tape manufacturers may try to avoid paying the levy by selling tapes of Argentinian dance music on which there is no copyright — the user could easily record over these.

There are other reasons why the proposal is bad. For the last 10 years, perhaps longer, the record industry has been producing consistently bad-quality pressings, distorted records, records with scratches and bumps on them from the moment they were pressed in low-quality vinyl. The quality of the products has been so poor that it has often seemed hardly worth the bother of buying a record, knowing that it would have to be returned. Because of the falling sales, the prices have been pushed up in an attempt to stay profitable, thereby trapping the industry into a circle of decline.

Now, rather than improve their own quality control, the companies look to a levy, raising extra cash from the wide section of the population who use blank tapes quite legally. We think this is wrong and would like your help to campaign against it. Please either write to your MP, or your local or national newspaper. Or else write to *Your Computer* and we will forward copies of your letters on to a number of MPs who will be taking an interest in opposing this proposal. If your response is large enough we might be able to stop the proposal getting off the ground.

THE NEXT ISSUE of Your Computer *will cost 60p and the price of a subscription will rise from £6 to £8. If you want to subscribe to* Your Computer, *do it before the end of the year and you save £2. There is a subscription card between pages 86 and 87 of this magazine.* ■

MicroValue

MONEY SAVERS

with the exclusive MicroValue 12 Months Warranty*

MicroValue Exclusive

MV1 - a Z80 computer for £105 + VAT

The MV1 computer kit uses the ubiquitous Nascom 1 Pcb and the Z80 CPU. Interfaces are included for television, printer and cassette. 2K memory, Gemini power supply (drives up to 3 extra boards). Cherry full ASCII keyboard and Quantum Graphics are also included. Available with either an ASCII version of the Nas-Sys 3 monitor, or a Tiny BASIC. MV1 is expandable to Gemini 80-BUS specification.

SAVE £107.50

MicroValue Exclusive

MicroValue's 'Nascom Special'

SAVE OVER £65

We've put together a microcomputer kit containing the Nascom 2, Nas-Sys 3, Graphics ROM, Bits & P.C.'s programmers aid, Gemini 3 APSU, 16K RAM Board and mini motherboard. The result is a powerful micro using market proven boards and components.

RRP OVER £405 + VAT
MicroValue price
£340 + VAT

MicroValue Exclusive

SHARP MZ80K with Super Graphics

SAVE £200!

The 48K RAM System is offered at a rock bottom price with the Quantum Micros Hi Res Graphics which gives resolution down to a single dot and high res. plotting. Characters are user definable and the pixel characters actually *join*. Five free games packages are included too!

RRP £645 + VAT
MicroValue price
£445 + VAT



SHARP

£30 worth of accessories FREE with every Epson Printer

	MicroValue price
Epson MX80T	£359 + VAT
Epson MX80FT1	£399 + VAT
Epson MX80FT2	£465 + VAT
Epson MX100	£575 + VAT

Buy one of the above Epsons from MicroValue and we'll give you a Pack of Fanfold paper, Spare Ribbon Cartridge, Interfacing Document and Connecting Cord for Multiboard or Nascom. The accessories are worth £30 but you can have them absolutely FREE.



Cheapest Printer in the UK!

Nascom IMP + Graphics Only £199 + VAT

SAVE £156

MicroValue has slashed the price of the 80cps, 80 column IMP dot matrix printer. And added Imprint's high res. graphics and double width character option. IMP has bi-directional printing and friction/tractor feed.

RRP £355 + VAT
MicroValue price
£199 + VAT

NASBUS Compatible DOUBLE DENSITY Disk System - Available Ex Stock

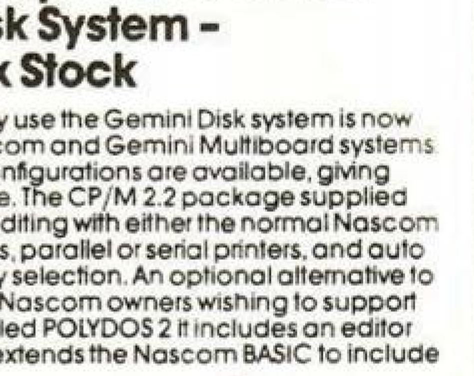
With hundreds in daily use the Gemini Disk system is now the standard for Nascom and Gemini Multiboard systems. Single or twin drive configurations are available, giving 350K storage per drive. The CP/M 2.2 package supplied supports on-screen editing with either the normal Nascom or Gemini IVC screens, parallel or serial printers, and auto single-double density selection. An optional alternative to CP/M is available for Nascom owners wishing to support existing software. Called POLYDOS 2 it includes an editor and assembler and extends the Nascom BASIC to include disk commands.

Single drive system
(G809, G815/1)
£465 + VAT

Double drive system
(G809, G815/2)
£690 + VAT

CP/M 2.2 package
(G513)
£100 + VAT

Polydos 2
£90 + VAT



MicroValue

MONEY SAVERS

with the exclusive MicroValue 12 Months Warranty*

MicroValue Exclusive

MV2-Twin Z80A Controlled Development Computer

The fully built and tested MV2 microcomputer is controlled by two Z80A microprocessors. Interfaces include RS232, cassette, 2 × 8 bit parallel ports, and graphics including programmable graphics. It provides 80 × 25 screen format and includes 64K RAM, Integral PSU and full ASCII keyboard.

Software written to run under the RP/M ROM based monitor can be transferred to disk to run under CP/M at a later date. This rugged computer is ideal for educational and industrial environments and is supplied with the advanced COMAL structured BASIC.



SAVE £76.50

MicroValue price

£595 + VAT

MicroValue Exclusive

MV3-Low Cost Business System

SAVE £91.50

A complete, fully built, double disk based CP/M version of MV2 system. Supplied with VDU and keyboard. Full CP/M software library available.

The MV3 is a highly reliable system of modular internal construction backed by the full MicroValue warranty.

MicroValue price only

£1550 + VAT

MicroValue Exclusive

MV3 WP System Word Processing System for only £2550 + VAT

By combining the power of the MV3 and the popular CP/M word processing package wordstar, and then adding the Olympia daisy-wheel printer, we can now offer a system for computerised letter and report writing, catalogue and price list compilation, etc., etc., as well as handling all the data processing functions of the MV3.



MicroValue Business Software Package

A fully integrated business package is available for the MV3 and MV3WP systems and includes Sales, Purchase, and Nominal ledgers and Stock Control, plus all documentation. MicroValue price – **£400** + VAT

MicroValue Exclusive

New Software for Nascom Systems

POLYDOS 1 A disk operating system for use with Nascom 1 or 2 and Gemini G805 Disk Systems. An incomparable and extremely well presented DOS that includes an editor and assembler and adds disk commands to the Nascom BASIC. MicroValue price **£90 + VAT**

MATHSPAK Double precision maths package on tape. MicroValue price – **£13 + VAT**

MATHSPAK Handler Used in conjunction with MATHSPAK. MicroValue price – **£9.95 + VAT**

Command Extender For use with MATHSPAK it extends BASIC's reserve word list. MicroValue price **£9.95 + VAT**

Logic Soft Relocater An integrated assembler and disassembler package which allows disassembly and reassembly from anywhere on the memory map. MicroValue price – **£13 + VAT**

SAVE MORE MONEY

Standard Firmware for Nascom at Reduced prices

NASPEN	RRP £30 + VAT	**MicroValue price £20 + VAT**
Nas-Sys 3	RRP £25 + VAT	**MicroValue price £20 + VAT**
NasDis - D-Bug (EPROM)	RRP £50 + VAT	**MicroValue price £30 + VAT**
NasDis - D-Bug (TAPE)	RRP £40 + VAT	**MicroValue price £20 + VAT**
Imprint	RRP £30 + VAT	**MicroValue price £20 + VAT**
Bits & PCs Prog. Aid	£28 + VAT	**MicroValue price £20 + VAT**

80 x 25 Video for Nascom

Nascom owners can now have a professional 80 x 25 Video display by using the Gemini G812 Intelligent Video Card with onboard Z80A. This card does not occupy system memory space and provides over 50 user controllable functions including prog character set, fully compatible with Gemini G805 and G815/809 Disk systems. Built and tested **£140** + VAT

*MicroValue Warranty

All products, except kits, sold by MicroValue dealers are supplied with 12 months' warranty and will be replaced or repaired by any dealer (even if you didn't buy it from him) in the group in the event of faulty manufacture.

YOUR LOCAL MICROVALUE DEALER

All the products on these two pages are available while stocks last from the MicroValue dealers listed on right. (Mail order enquiries should telephone for delivery dates and post and packing costs.) Access and Barclaycard welcome.

BITS & PC'S
4 Westgate, Wetherby, W. Yorks.
Tel: (0937) 63774.

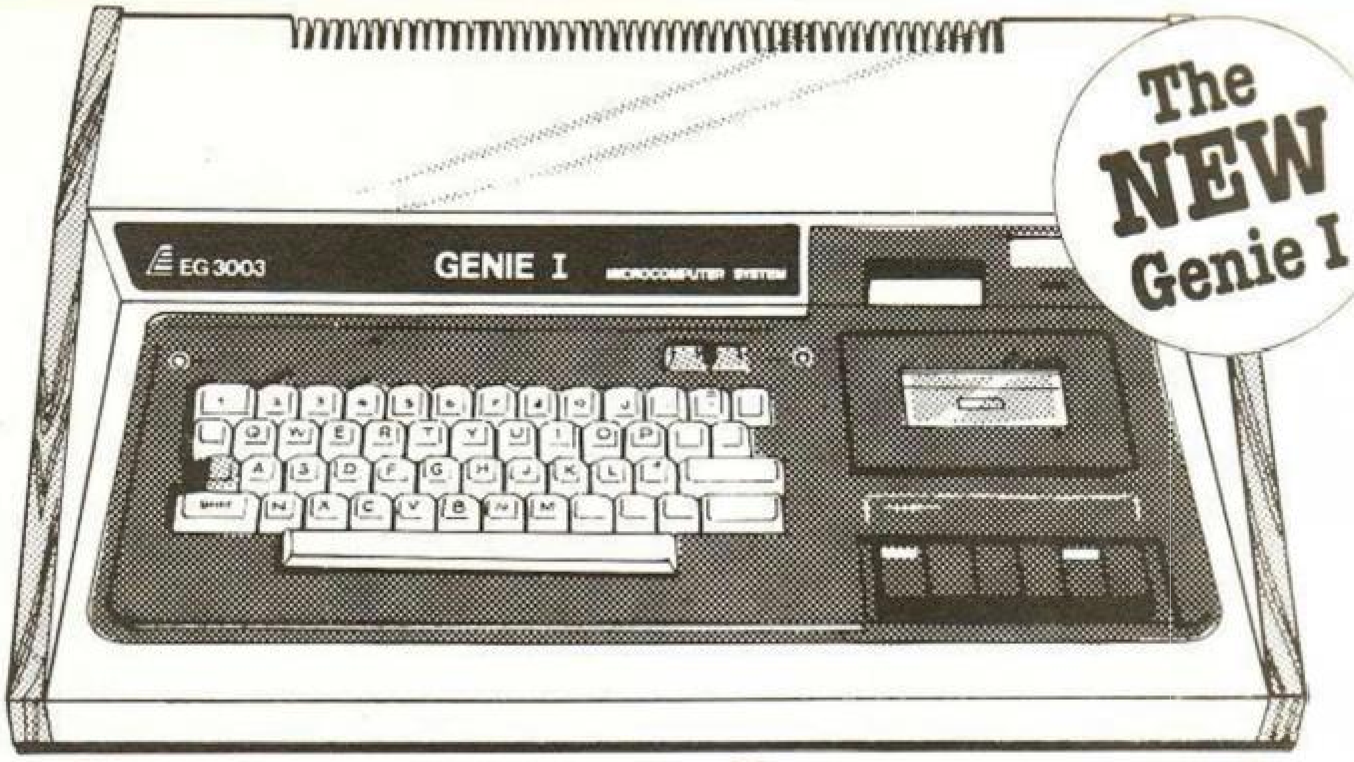
ELECTROVALUE LTD.
700 Burnage Lane, Burnage, Manchester M19 1NA.
Tel: (061) 431 4866.

28 St Judes, Englefield Green, Egham, Surrey TW20 0HB.
Tel: (0784) 33603. Tlx: 264475.

TARGET ELECTRONICS
16 Cherry Lane, Bristol BS1 3NG.
Tel: (0272) 421196.

INTERFACE COMPONENTS LTD.
Oakfield Corner, Sycamore Road, Amersham, Bucks.
Tel: (02403) 22307. Tlx: 837788.

HENRY'S RADIO
404 Edgware Road, London W2.
Tel: (01) 402 6822.
Tlx: 262284 (quote ref: 1400).



The NEW Genie I

ingenious!

... that's the only word to really describe the superb Genie microcomputer system, the home computer which is compatible with the TRS 80, and ideal for all micro – enthusiasts, especially the committed hobbyist.

Genie has now been upgraded to Genie I, incorporating all of the original, excellent features, but with the addition of:

- Extended BASIC, including RENUMBER and SCREEN PRINT.
- Full upper and lower case, flashing cursor and auto-repeat on all keys.
- An internal SOUND UNIT, to add a new dimension to your own programs.
- A MACHINE LANGUAGE MONITOR, with Display, modify, enter and execute (with break points) facilities.

Genie I has all of this, plus the built-in cassette deck, 16K RAM, 12K ROM with BASIC interpreter, full-size keyboard, an extremely wide range of new and up-dated peripherals, and literally 1000's of pre-recorded programmes available.

Yet, almost unbelievably, the price of Genie I is even lower than that of the original Genie!

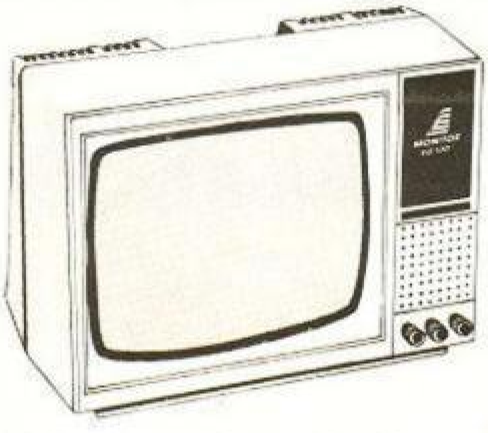
Ingenious for business

The Genie II is a major breakthrough for small business computers. Harnessing all the advantages of Genie I, including low price, Genie II adapts perfectly to commercial functions with the following features:

- Numeric keyboard
- Four usable, definable function keys
- Extension to BASIC
- Basic business commands
- Fully expandable with the same periperals

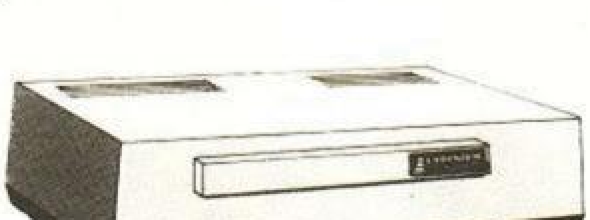


LOWE electronics



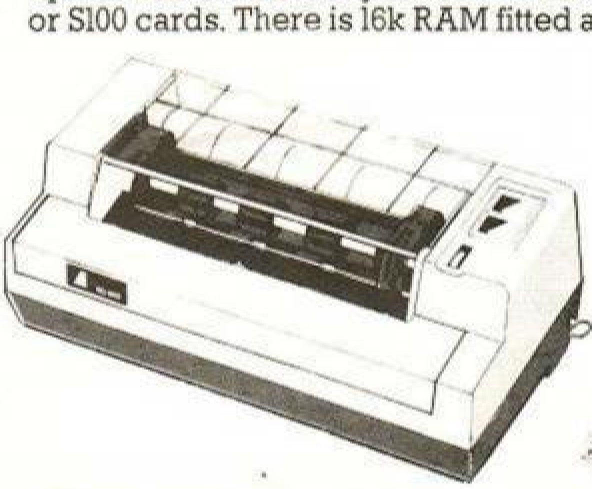
New . . . 12" Monitor.

There is now a choice of 2, 12" monitors with the Genie I system, allowing a clear, easy to read image, and no interference with your domestic T.V. viewing. The new EG 101 comes with an updated, green phospher tube.



New! . . . Expander Box.

An updated Expansion Box (EG 3014) is a major feature of the new Genie I system, and unleashes all its possibilities, allowing for up to 4 disk drives with optional double density. It connects to a printer, or RS232 interface or S100 cards. There is 16k RAM fitted and it has a new low price!



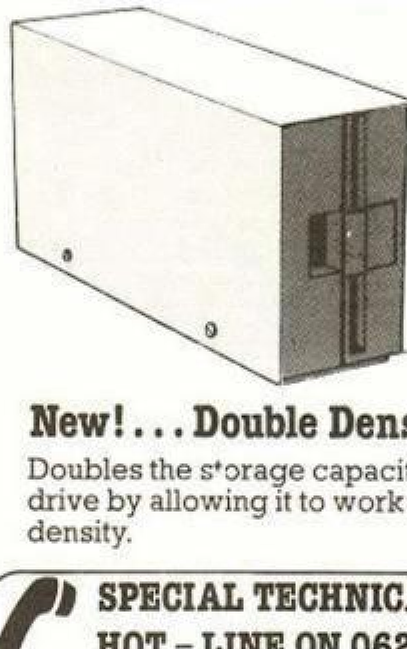
New! . . . Printer

The EG 602 printer can be connected to the Genie either through the expander, or directly into the computer using the Parallel printer interface. It is a compact unit, with an 80 column, 5 x 7 matrix print-out, operating quietly and efficiently at 30 characters per second.



New! . . . Parallel Printer Interface.

Enables you to connect the printer directly into the Genie computer without using the expansion box.

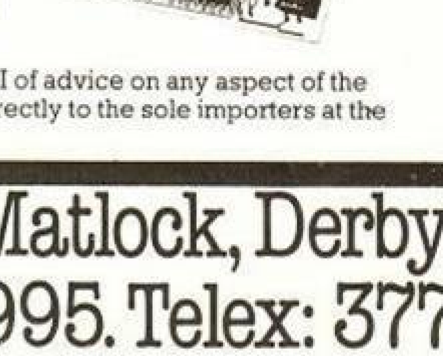


Disk Drive.

As well as the obvious advantage of mass storage, the addition of the disk system to the Genie means much faster access to other languages and full random access file handling. Up to 4 of these 40 track drives can be used on a system.

New! . . . Double Density Adaptor

Doubles the s'orage capacity of your disk drive by allowing it to work double - density.



SPECIAL TECHNICAL GENIE HOT – LINE ON 0629 4995

for all your technical advice and service back-up on any aspect of the Genie system direct from the experts!

For full details and demonstration of Genie I, Genie II of advice on any aspect of the system, either call in to your local dealer, or write directly to the sole importers at the address below.

Genie I and II approved dealers

AVON Microstyle, Bath, 0225 334659/319705. **BEDFORD** Computopia, Leighton Buzzard, 0525 376600. Comserve, Bedford, 0234 216749. **BERKSHIRE** P.C.P., Reading, 0734 589249. **BIRMINGHAM** Laskys Microdigital, Birmingham, 021-632 6303. Ward Electronics, Birmingham, 021-554 0708. **BRISTOL** Laskys Microdigital, Bristol 0272 20421. **BUCKINGHAMSHIRE** Photo Acoustics, Newport Pagnell, 0908 610625. Interface Components, Amersham, 02403 22307. **CAMBRIDGESHIRE** Cambridge Micro Computers, Cambridge, 0223 314666. **CHESHIRE** Hewart Electronics, Macclesfield, 0625 22030. Mid Shires Computer Centre, Crewe. Laskys Microdigital, Chester, 0244 317667. **CUMBRIA** Kendall Computer Centre, Kendall, 0539 22559. Northrock Music, Carlisle, 0228 37114. **DERBYSHIRE** Kays Electronics, Chesterfield, 0246 31696. T Crossley, Chesterfield, 0246 850357. **DORSET** Blandford Computers, Blandford Forum, 0258 53737. Parkstone Electrics, Poole, 0202 746555. **ESSEX** Emprise, Colchester, 0206 865926. Compuskill, Romford, 0708 751906. Infolab, Chelmsford, 0245 357111. Micro Computer Services, Clacton on Sea, 0255 29018. CSSC, Ilford, 01-554 3344. **GLOUCESTERSHIRE** MPL Computers, Cheltenham, 0242 582090. Petrie Systems, Cheltenham, 0242 584060. Computer Shack, Cheltenham, 0242 584343. Zeta Computers, Stonehouse, 045 382 2444. **HERTFORDSHIRE** Photo Acoustics, Watford, 0923 40698. Watford Electronics, Watford, 0923 40588/37774. Q Tek Systems, Stevenage, 0438 65385. Comp Shop, New Barnett, 01-441 2922. **KENT** Matrix Computer Systems, Beckenham, 01-685 7508/7551. Business Systems, Hempstead, 0635 362652. The Computer Room, Tunbridge Wells, 0892 41645. SMG Microcomputers, Gravesend, 0474 55813. Swanley Electronics, Swanley, 0322 64851. **LANCASHIRE** Laskys Microdigital, Liverpool, 051-227 2535. Mighty Micro, Burnley, 0282 58758. Leisuronics, Blackpool, 0253 27590. Harden Microsystems, Blackpool, 0253 27590. Micro Chip Shop, Fleetwood, 03917 79480. Sound Service, Burnley, 0282 38481. Computercat, Leigh, 0942 605730. Laskys Microdigital, Preston, 0772 59264. **LEICESTERSHIRE** Eley Electronics, Leicester, 0533 871522. Arden Data Processing, Leicester, 0533 22255. Kram Electronics, Leicester, 0533 27556. **LONDON (CENTRAL)** City Microsystems, EC2, 01-588 7272/4. **LONDON (NORTH)** Radio Shack, NW6, 01-624 7174. Comp Shop, Edgeware Road, 01-262 0837. Chromasonic Electronics, N19, 01-263 9493. Wason Microchip, N18, 01-807 1757/2230. Comp Shop, New Barnet, 01-441 2922. **LONDON (WEST)** Henry's Radio, W2, 01-402 6822. BDM Computer Marketing, W9, 01-286 7374. **LONDON (SOUTH)** Laskys Microdigital, Kingston, 01-546 1271. **MANCHESTER** Laskys Microdigital, Manchester, 061-832 6087. ABC Supplies, Levenshulme, 061-431 9265. **NORTH EAST** 3 Line Computing, Hull, 0482 859169. Derwent Radio, Scarborough, 0723 65996. Briers Computer Services, Middlesborough, 0642 242017. General Northern Microcomputers, Hartlepool, 0783 863871. HCCS Associates, Gateshead, 0632 821924. **NORTHANTS** Arden Data Processing, Peterborough, 0733 49577. **NOTTINGHAMSHIRE** University Radio, Nottingham, 0602 45466. Midland Microcomputers, Nottingham, 0602 298281. Laskys Microdigital, Nottingham, 0602 415150. Mansfield Computers, Mansfield, 0623 31202. East Midland Computer Services, Arnold, 0602 267079. Electronic Servicing Co., Lenton, 0602 783938. **NORFOLK** Anglia Computer Centre, Norwich, 0603 29652. Bennetts, Dereham, 0362 2488/9. **OXFORDSHIRE** Magnus Microcomputers, Kidlington, 08675 6703. Micro Business Systems, Whitney, 0993 73145. **SCOTLAND** Computer and Chips, St Andrews, 0334 72569. Laskys Microdigital, Edinburgh, 031-556 2914. Scotbyte Computers, Edinburgh, 031-343 1005. Laskys Microdigital, Glasgow, 041-226 3349. Esco Computing, Glasgow, 041-204 1811. Silicon Centre, Edinburgh, 031-332 5277. **SHROPSHIRE** Tarrant Electronics, Newport, 0952 812134. **SOUTH** Aerco Gemsoft, Woking, 04862 22881. Castle Electronics, Hastings, 0424 437875. Gamer, Brighton, 0273 69824. **SOUTH WEST** Diskwise Ltd, Plymouth, 0752 276000. Diskwise Ltd, Callington, 05793 3780. Electrosure, Exeter, 0392 56280/56687. West Devon Electronics, Yelverton, 082 285 3434. **SUFFOLK** Rebvale Computers, Bury St Edmunds, 095 381 316. Marshion Electronics, Ipswich, 0473 75476. Microtek, Ipswich, 0473 50152. Elgelec Ltd, Ipswich, 0473 711164. **SURREY** Croydon Computer Centre, Thornton Heath, 01-689 1280. Catronics Ltd, Wallington, 01-667 6700/1. **SUSSEX** Nestra Electronics, Chichester, 0243 512861. **WALES** Morriston Computer Centre, Swansea, 0792 795817. MRS Communications, Cardiff, 0222 616936/7. Tryfan Computers, Bangor, 0248 52042. **WEST MIDLANDS** Allen TV Services, Stoke on Trent, 0782 616929. Microprint, Stoke on Trent, 0782 48348. **WILTSHIRE** Everyman Computers, Westbury, 0373 823764. **YORKSHIRE** Advance TV Services, Shipley, 0274 585333. Amateur Radio Shop, Huddersfield, 0484 20774. Thomas Wright, Bradford, 0274 663471. Scene and Heard, Halifax, 0422 59116. Spot Computer Systems, Doncaster, 0302 25119. Superior Systems Ltd, Sheffield, 0742 755005. Laskys Microdigital, Sheffield, 0742 750971. Photo Electrics, Sheffield, 0742 53865. **EIRE** Compshop, Dublin, Dublin 74933. **NORTHERN IRELAND** Business Electronic Equipment, Belfast, 0232 46161. Brittain Laboratories Ltd, Belfast, 0232 28374. **CHANNEL ISLANDS** GB Organs, St Savior, 0534 26788/23564

Chesterfield Road, Matlock, Derbyshire DE4 5LE
Telephone: 0629 4995. Telex: 377482 Lowlec G.

YOUR LETTERS

ZX-81 CODE

After reading the November issue from cover to cover, I tried to run the programs in listing 2 of Brendan Clancy's informative article. It seems to omit four values between addresses 16586 and 16661, and two values between addresses 16686 and 16736.

This was my first attempt at machine-code programming, so I took great care to enter all the values and have re-counted the number of bytes given in the listing several times. Please can you print the missing values, or tell me where I've gone wrong, so that I can at last see the advantages of machine-code programing?

Nick Carter,
Milford-on-Sea,
Hampshire.

■ **Before byte number 16661 insert two lines which should read:**

LD HL 16438 33,54,64; crash
INC(HL) 52

Just before byte number 16736 insert:

JR Z 6 40,6

Line number 2 in the first Basic program should read line number 5, and there should be a space between the " " in line 20 in the second Basic program.

FAULTY CASSETTE

The article by Eric Deeson on ZX programs in the October issue was excellent. It arrived in time to stop me from pulling the remaining grey hairs from my head. I had repeatedly tried to load my cassette and had tried everything, including prayer, without success. The article reassured me and I have subsequently returned the cassette to Sinclair Research for exchange.

I should add that the cassette sent was a free cassette issued to calm my nerves because I had waited since July 11 for a RAM. I have not as yet received two cassettes ordered at the same time. What a pity that such an excellent machine as the ZX-81 seems to be marred by such inadequate software and after-sales service. I hate to think what use the cassettes will be when they eventually arrive.

D B Orpin,
Binbrook,
Lincoln.

RAM ALTERNATIVE

In the October issue of *Your Computer,* P C Jowsey indicated in Response Frame that the circuit diagram supplied with the kit version of the ZX-81 suggested the use of a 4816 2K RAM instead of the 1K RAM supplied.

I have carried out this modification but using a 4802 which is an updated 24-pin version of the 4816. It is pin-compatible with the 4118 except that pin 19 must go to the A10 rail — as for the 4816.

The 4864 could be fitted to replace either of the devices mentioned — this will give 8K of RAM. All these RAMs are available in the Mostek Bytewide series and can be fitted without any serious modifications to the printed-circuit board.

I am a newcomer to the Sinclair ZX-81 which I have just finished constructing. I read with interest that some users have had problems recording on cassettes from the ZX-81.

Some very cheap cassette machines have an integration capacitor across the speaker or earphone socket. Mine had one of 0.068μF and I found that it upset the replay into the computer. Removal of this capacitor completely solved the problem. If you are lucky enough to have a circuit diagram, check to see if your recorder has this capacitor.

I found that cheap cassettes could not be used. Their oxide is such that they do not saturate properly, and recording digital information becomes impossible. Medium-quality cassettes such as BASF SL or similar are suitable.

I hope this helps others who may have problems with the ZX-81 — it is possible that the problems do not lie within the computer but with the cassette recorder.

Stuart Stirling,
Ilford,
Essex.

ZX-81 CONTACTS

I have found the solution to fitting the ZX-81 RAM pack securely. The main problem is that the RAM pack wobbles when nudged even slightly, which is due to poor contacts and results in total data loss.

To cure the problem permanently, you must first remove the ZX circuit board from the casing and then plug in a soldering iron. When the iron is hot, apply small quantities of solder to the strips of solder already present on the board. This should be done on both sides of the output port, so that the RAM pack fits much more tightly.

I also noticed, in your October edition, that Eric Deeson had loading problems with some of the software he was reviewing. He blames software-recording quality, but I am reasonably sure this is not the cause. The problems are more likely caused by cassette-player incompatibility.

I have found that speeds and tones vary according to the cassette players used — I have eight of them. The simplest cure for this is not to send the software back to the suppliers but to borrow someone else's player and try with that. When the program is loaded into the computer, save it again on your own cassette player.

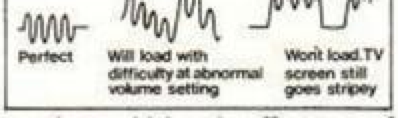
Jem Software,
Lilbourne,
Warwickshire.

SCOPE TEST

After a look through the *Your Computer* survey of program and tape standards in the October issue, I think it is worth mentioning that an oscilloscope is extremely useful when making or testing ZX-81 tapes.

So far, we have only examined our own tapes — made in the course of developing a reliable mass-duplication system.

The oscilloscope is probably the best tool for distinguishing between perfect tapes, those that will load with a tail wind and some luck and those not worth spending time on. The only problems that the oscilloscope will not bring to light quickly



are those which only affect part of the tape, e.g., dropouts and bad patches. These can be very largely avoided in production by using good-quality tape.

All our batches of tapes are tested by sampling. If a few samples load impeccably, and the wave-form is near perfect, we accept the batch. If anything suspicious is found, we test part of each tape for a few seconds on the oscilloscope.

Mike Salem,
Hilderbray Ltd,
London NW1.

"LIFE" ON 16K

ZX-81 owners who have tried to run Sinclair's Life program with a 16K RAM pack fitted will know that the system crashes — albeit gracefully. The following procedure permits the program to run with 16K. Load the program and enter these three lines:

```
900 FOR N=16666 TO
    16550 STEP -1
910 POKE (N+2), PEEK N
920 NET N
```

Then type:

```
GO TO 900
```

Amend line 700 to

```
700 LET C=USR 16572
```

Lines 500 to 540 inclusive are now redundant and lines 900 to 920 should be deleted. Save the program before running.

This procedure works because the original program ran with a page width of 16 characters, whereas the 16K RAM creates a 32-character display. The two additional memory locations created by opening the machine code at the right place contain the shift left instruction, and effectively multiply the page width by two.

Martin Buckley,
Southampton.

SKETCH SUCCESS

I agree whole-heartedly with the opinion G A Bobker expressed in the August/September issue about Sinclair cassettes. Admittedly it requires very little work to make them run smoothly, but why leave it to the poor purchaser — surely they could have been altered before release? I was undoubtedly unlucky — I bought cassettes 1 and 4 and have had to return number 4 because of errors.

Eric Deeson's Sketch Pad program in the same issue proved a great success with my children, but they kept entering too few or too many characters, which of course terminates the program. This can be avoided by adding between lines 9 and 10 or 90 and 100.

```
95 IF LEN A$< >3 THEN GO TO 60
```

Gillian Turner,
St Neots,
Cambridgeshire.

VIC OR ATOM?

I am at present considering buying a microcomputer. I read the September issue of *Your Computer* and found it very informative. However, having joined the local computer club — something I would advise anyone in a similar position to do — I have had a chance to use several home computers and I am forced to disagree with your views on the Vic.

At the price of about £300 for the Vic-20, the Super Expander Cartridge and the cassette recorder, I cannot see that it is good value when compared to the expanded Acorn Atom. This machine offers twice the screen capacity and if the colour card is used, it is more than a match for the Vic graphics.

Admittedly, the Acorn has a non-standard Basic, but once you become accustomed to it, aided by the excellent manual, it is easier to understand than the Pet version. Added to this is the fact that the Atom comes as standard equipped with an assembler, rather than the Vic's clumsy Peek and Poke — surely a big point in its favour.

In short I feel that this British competition is more than a match for imported machines. I would advise anyone wanting a computer to buy British — not only out of patriotism but for value for money.

N Goodwin,
Horley,
Surrey. ■

Co-op to help U.K. clubs

CO-OPERATIVE Retail Services is to create a network of home-computer clubs for its customers, members and staff. It has long been the practice of the Co-operative movement to assign part of its profits to financing educational, cultural and leisure-time schemes. In the past, the money has been spent on a wide range of activities including choirs, youth groups and classes.

The Co-operative Societies' national member relations officer, Frank Dent, says that he sees home computing as possibly the biggest leisure growth area which could soon rival photography. "In the Co-op, we have always tried to respond to new needs. I am convinced that there must be many thousands of people just waiting to join home-computer clubs".

To start such a club, contact Frank Dent, CRS Ltd, 29 Dantzic Street, Manchester M4 4BA.

Upgrade ZX-81 memory to 48K

OWNERS OF either the Sinclair ZX-80 or ZX-81 microcomputers are constantly frustrated by the shortage of user-memory available for their machines. The meagre amounts of RAM available with the basic machines is enough for a very simple program but for one of any degree of sophistication, the low-capacity memory is prohibitive.

Many ZX fans find the greed for memory satisfied by the 16K RAM packs which wobble about on the rear connector. Yet for the more avaricious memory user, even a generous 16K may not be enough. For those, Memotech has introduced a new 48K memory extension.

The Memotech memory-extension board will allow the ZX-81 to run 48K Basic programs which can include up to 16K of assembly code. The memory is available in either a ready-built form or, for the more adventurous, a kit is available. Unlike the 16K RAM pack, the 48K memory extension is complete with a power supply which services the computer as well as the memory extension.

The memory extension resides in a case on which the microcomputer sits. It has a fully-buffered control-data-address bus with a printed-circuit board 40-way header plug.

The memory is configured in such a way that there is a 16K "gap" between 16K and 32K for assembly programming. Top of memory can then be set at any point up to 64K.

The ZX memory expansion costs £129 plus VAT built, or £109 plus VAT for the kit. For the Nascom 2 the memory expansion costs £85 plus VAT. Memotech: 0865 513356.

Sinclair seals Japan deal



CLIVE SINCLAIR has good reason to look pleased with himself. The two gentlemen with him are British representatives of Mitsui, the giant Japanese trading company, who are about to begin marketing the Sinclair ZX-81 microcomputer in Japan.

M Ohtaki, left, assistant general manager at the London branch of the Mitsui organisation and Hiroshi Shimizu, right, the manager of Mitsui computers, told *Your Computer* that Mitsui would be selling the machines in Japan using the same mail-order techniques Clive Sinclair pioneered in the U.K.

"We will retain the English-language keyboard — the difference in languages will not be important in the market where we are selling. We regard the ZX-81 as an educational toy".

Shimizu is the man responsible for negotiating the deal which has opened up the world's second biggest mircocomputer market to the world's best-selling microcomputer.

The announcement was made simultaneously in London and Tokyo, and is of a great deal of importance as Mitsui is responsible for 10 percent of all imports to Japan — everything from British Leyland cars to Scotch whisky. Mitsui expresses a wish to import more goods to the land of the rising sun and in fact extended an open invitation to British innovators to contact the company to sell any high-technology product.

Clive Sinclair said that he originally believed the Japanese market to be too difficult to enter. However, with the expertise and resources of one of the world's largest trading companies behind him, he said he was confident.

Payroll geared to small firm

A SMALL-FIRM payroll program together with step-by-step instructions, which runs on the ZX-81 with 16K RAM pack and printer, will perform all the payment and deduction calculations and keep the records for a small company with 30 or fewer employees.

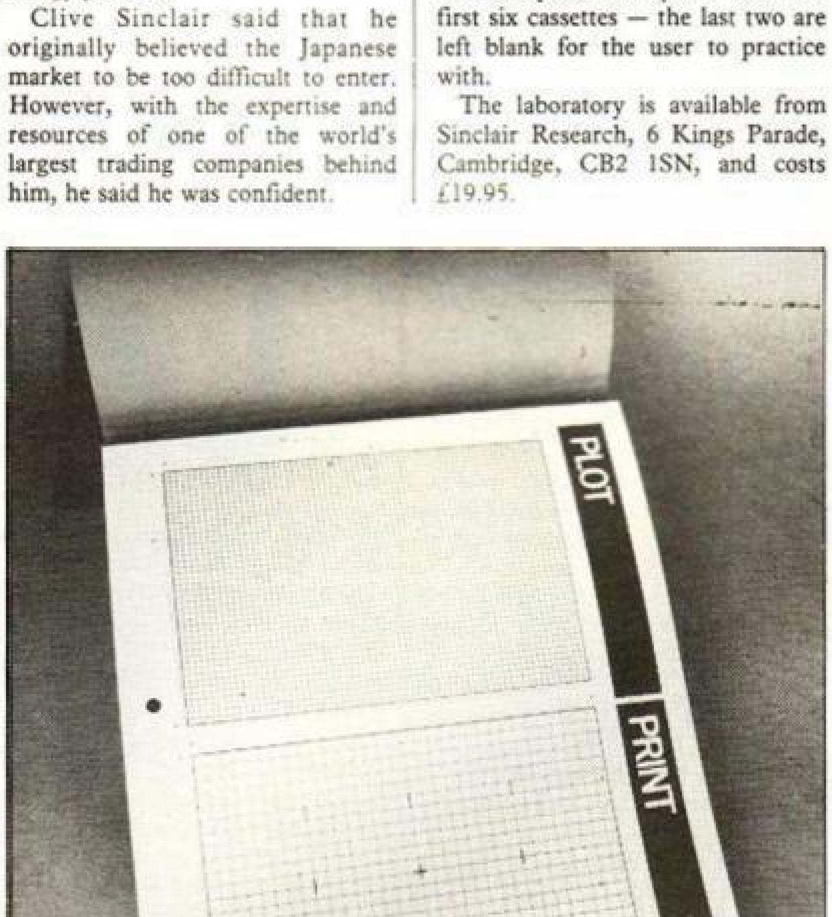
The program can also cope with bonuses and any occasional payments which need to be made. It can also gross a nett payment. There is a program-replacement service in case of any tax changes. Contact Hilderbray Ltd, 8-10 Parkway, London NW1 7AA. Telephone 01-485 1059.

Learning Basic in the lab

SINCLAIR Research has developed a hands-on ZX learning laboratory to enable users to learn programming at the machine. Developed by Sinclair to meet popular demand, the laboratory comprises eight cassettes and a 160-page manual.. The 20 programs each demonstrate an aspect of ZX-81 programming. These aspects are spread over the first six cassettes — the last two are left blank for the user to practice with.

The laboratory is available from Sinclair Research, 6 Kings Parade, Cambridge, CB2 1SN, and costs £19.95.

This is the ZX-81 Print 'n' Plotter Jotter from Butler, Currie and Hook. It is a useful aid for anyone interested in exploiting the graphics capabilities of the Sinclair microcomputers to the full. Consisting of a tear-off pad of 100 leaves, the Jotter has each page printed with a ZX print grid and a ZX plot grid. Each grid is fully numbered and clearly definable. The pad measures 11.75in. by 8.25in. — the standard A4 size. On the print grid each of the 704 character positions are shown and numbered. The plot grid has all 2,816 pixel co-ordinates. The price of the ZX-80/81 Print 'n' Plotter Jotter is £3.50 for one pad. When ordering by post, second and subsequent pads cost £3.15; if five are ordered, a free ZX Print 'n' Plotter Film is yours. The Film costs £2 if bought separately. Discounts on larger quantities are offered to clubs and user groups. The retailer of the Jotter — Butler, Currie and Hook, 19 Borough High Street, London SE1 — is planning to extend the scope of the Jotters to cover other popular microcomputers. Butler, Currie and Hook can be contacted by telephone on 01-403 6644.



Schools enjoy special price

MORE THAN 2,300 secondary schools have opted for the Sinclair ZX-81. The machines were sold to the schools under a special low-price scheme which was run earlier this year by Sinclair in conjunction with educational distributors Griffin and George. The scheme was Clive Sinclair's personal bid to widen the choice of microcomputer equipment available to schools.

The Government-assisted scheme restricts the choice of machines to either the Research Machines 380-Z or the new BBC computer from Acorn. Clive Sinclair commented that "although we welcome the Government's initiative, we felt that it did not fully account for the needs of all schools. We believe that the success of our scheme vindicates our approach as both practical and economic".



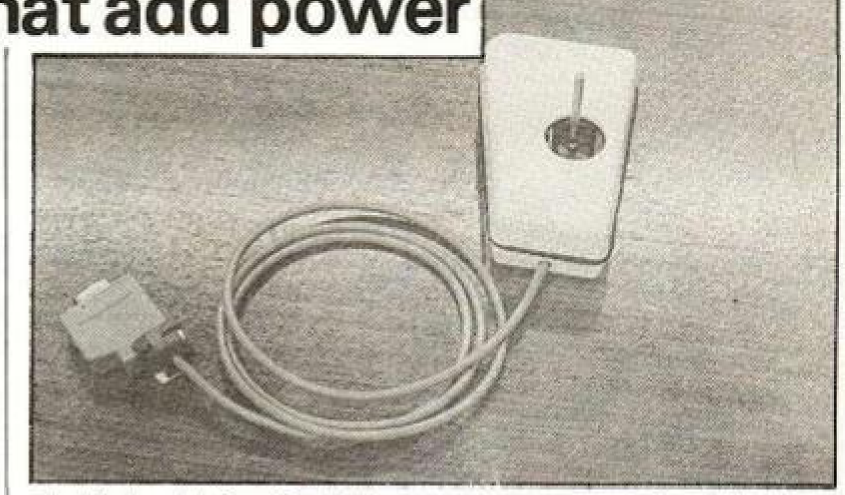
The first ZX Microfair was such an overwhelming success that people queued all the way around London's Central Hall, Westminster, for hours in the pouring rain. Considerably more than 5,000 ZX enthusiasts attended and the opening hours had to be extended so that everyone could have a chance to see the exhibits. The 50 stands in the packed hall reported extremely brisk business, with some stands recording sales well into four figures. Because of the response, organiser Mike Johnston is planning a second fair to be held at Central Hall, Westminster, on Saturday January 30. This time, to cope with the rush, the doors will be open from 10.30 am until 8.30 pm. The floor space will be doubled to give everyone room to breath. Anyone interested in exhibiting should contact Mike Johnston — after 7pm, on 01-801 9172.

Vic extras that add power

A WIDE RANGE of Commodore-approved peripherals for use with the new CBM Vic-20 microcomputer has been released by Stack Computer Services. The peripherals give the Vic many of those facilities associated with larger or more expensive computers. Judging by the response Stack has already received, these products are destined to be very popular.

The most obvious add-on to any small microcomputer system — after the tape recorder — is extra memory. RAM is always at a premium inside any microcomputer; especially those which do not provide much as standard. The 3.5K of RAM provided with the Basic machine should be enough to keep most happy for a month or two, but soon the user will want more.

Stack markets a 3K memory add-on for £39 which has the advantage of moving Basic to the same memory space as on the Commodore Pet — that means programs written on the Pet can be transferred. The unit plugs into the Vic and acts as the memory port so you do not lose any of the machine's facilities. Another advantage of the device is that it offers high-resolution graphics.



The Vic joystick from Stack is just one of a new range of peripherals.

There are in fact a whole range of memory expansions to suit every user. They are available in both CMOS and NMOS and range in size and price from £11.50 for 1K up to £184 for 24K plus a switchable 3K. Another memory product of interest is the 19K memory expansion which is battery-protected so that programs remain even when the computer is switched off.

The £25 Vic light pen will work in both the normal and the high-resolution modes. The pen enables you to interact with the computer without using the keyboard.

A CBM Vic can be used as a remote terminal to a large mainframe computer or can control a printer or any of a large number of other peripherals by the RS-232 interface. There are two versions, one at £17.25 and the full implementation of the standard interface costs £49.

Other peripherals include joysticks, two of which can be used if the multiplexor is used, an adaptor cable, a switchable ROM unit and a toolkit ROM.

Stack Computer Services accepts orders over the telephone if you have a credit card. 051-933 5511. However most of the peripherals will be available from your local dealer.

High-resolution on Tangerines

HIGH-RESOLUTION graphics of 256-by-256 definition are now a reality for Microtan users. The programmable graphic generator, developed by the Tangerine Users' Group, follows in the footsteps of the EPROM programmer and also offers a reverse-video ASCII character set.

For more details of these products, contact the Tangerine Users' Group on 0202-294393.

Forth reviewed for Microtan

TANGERINE IS assessing a Forth compiler for the Microtan. Its release will most likely coincide with the proposed new disc unit which is also under development. Forth is a structured programming language ideally suited to microprocessors and has been hailed by many as the language of the future.

Unlike Comal or Pascal, Forth bears little resemblance to any form of Basic. One of its key features is the manner in which users can define their own key-words.

Tangerine has withdrawn the discount facilities enjoyed by the Tangerine Users' Group, commonly known as TUG. To continue to provide users with some degree of after-sales support of one kind or another, the new customer-support engineer Paul Kaufman will edit an owners' newsletter *The Tansoft Gazette*. Free copies of the first issue will be sent to all Tangerine users.

Noise-generation board fills the sound slot

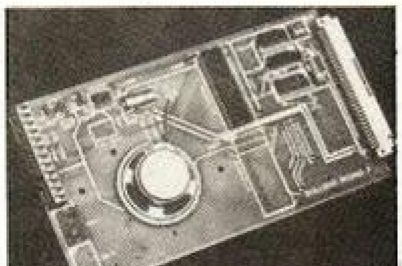
REDUNDANT mainframe computer designers do not die or fade away — they move into micros. At least that is what happened to some of those made redundant by ICL. They formed Bulldog Video Ltd, a company which will design and market products for the hobby and small-business microcomputer owner.

The first product to be released from the company is a programmable sound-generator board for the Tangerine microcomputer. It is Tangerine-sized, slots into the system motherboard and has an on-board speaker and an audio-amplifier so that sounds can be produced as soon as it has been inserted. For the perfectionist, three terminal blocks are provided so that it can be connected up to your Bang and Olufsen.

The board is centred on the General Instrument AY-3-8910, a powerful sound-synthesis chip. The sounds produced consist of a mixture of three channels, each with an independent tone generator and a white-noise generator. The amplitude of the sound can be managed by the envelope-control facility.

Documentation with the board explains how various sounds can be obtained. Bangs, gunshots, whistling bombs and explosions are there to be included in games. Music creation is also possible.

The sound board is programmed via a set of 14 registers which can be accessed from Basic, assembler or the Tanbug M instruction set. The board has full documentation, including the GI data manual and the Bulldog, a manual which contains sections on how to start, how to program the beast, a guide to building up a sound, notes on the physics of sound, a list of frequencies and how to obtain them and a note on installing external speakers.



The sound-generator box.

The board also contains two independent eight-bit input/output ports which can be used for control purposes. A second sound-producing chip can be added so that, for example, two chips could sing in harmony.

The Bulldog video sound-generator board costs £44.85, or £56.35 with two sound chips; both prices include VAT. Bulldog Video is at 52 Nash Square, Birmingham. Telephone 0299-266143.

AT LAST!

THE COMPLETE SINCLAIR ZX 81 BASIC COURSE

At last, a comprehensive text for your Sinclair ZX 81! The complete BASIC Course is a manual which will immediately become an indispensible work of reference for all your ZX 81 programming.

Whether you have never done any programming or whether you are an experienced microcomputer user, the Complete BASIC Course will provide itself to you as an invaluable aid.

The Complete BASIC Course is designed to teach you to write and develop BASIC programs for the Sinclair ZX 81 – no other books or aids are necessary. All is revealed in our easy step-by-step guide with programs and "test yourself" exercises all the way through.

As you become more proficient with computing, the Complete BASIC Course will continue to be an essential guide, giving you finger tip references, numerous advanced programming techniques and memory saving devices specifically for the Sinclair ZX 81.

HOW TO WRITE PROGRAMS:

Even if the idea of writing programs is completely mystifying to you, the Complete BASIC Course will show you just how easy it is. In no time you will be able to write and enjoy complex programs for whatever use you desire.

Using the proven "TOP-DOWN" approach, the Complete BASIC Course will show you systematic and simple ways to write programs. Even experienced programmers will benefit from this Course, making programs easier to write and less prone to error!

NUMEROUS EXAMPLES:

Every concept, every function is fully described by simple programs that you can enter on your Sinclair ZX 81 in minutes.

The Complete BASIC Course contains over 100 programs and examples! These programs illustrate the use and possibilities of the Sinclair ZX 81:

- Home use
- Financial analysis and planning
- Educational applications
- Games
- Mathematical applications
- Displays of 'Artificial Intelligence'

EVERY FUNCTION COVERED:

No matter what your application, what your confusion about any function, you will find it covered in the Complete BASIC Course.

A full and detailed discussion is included of even traditionally taboo topics such as USR, PEEK and POKE.

A handy alphabetical summary section lists all functions, and provides a short description and example programs of all topics.

A PERMANENT WORK OF REFERENCE:

The Complete BASIC Course is an excellent reference work for experienced programmers (including tips on using special techniques) as well as a comprehensive step-by-step guide for complete beginners.

The Complete BASIC Course has over 240 pages filled with information in an attractive durable ring binder – this is a lay-flat work of reference that deserves a place next to every Sinclair ZX 81 microcomputer.

OTHER TITLES AVAILABLE:

Melbourne House is the world's leading publisher of books and software for the Sinclair ZX 81.

The following titles are also available if you wish to expand your horizons:

BASIC Course Programs on Cassette -

All major programs in the BASIC Course are available pre-recorded in this set of cassettes. This is a valuable adjunct to the Course, saving you time and effort.

Not Only 30 Programs for the Sinclair ZX 81: 1K -

Not only over 30 programs, from arcade games to the final challenging Draughts playing program, which all fit into the unexpanded 1K Sinclair ZX 81 but also notes on how these programs were written and special tips! Great value!

Machine Language Programming Made Simple for the Sinclair -

A complete beginner's guide to the computer's own language – Z80 machine language. Machine language programs enable you to save on memory and typically give you programs than run 10-30 times faster than BASIC programs.

Understanding Your ZX 81 ROM -

A brilliant guide for more experienced programmers by Dr. Ian Logan, this book illustrates the Sinclair's own operating system and how you can use it. Includes special section on how to use machine code routines in your BASIC p+rograms.



Order Form:

Orders to Melbourne House Publishers 131 Trafalgar Road, Greenwich London SE10

(Correspondence to Glebe Cottage, Station Road, Cheddington, Leighton Buzzard, BEDS LU7 7NA)

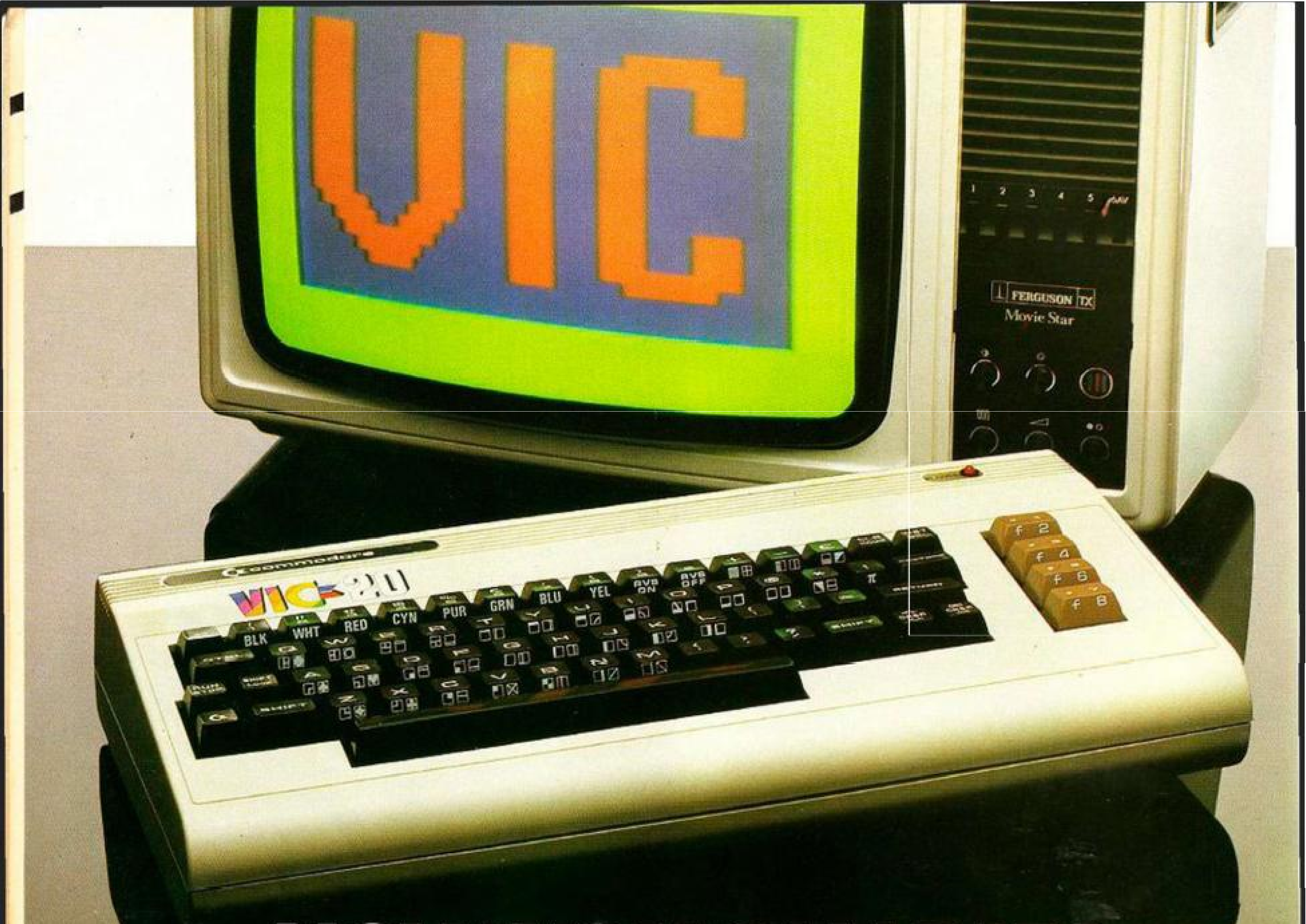
NAME:..

ADDRESS:..

....

....Postcode........................

The Complete Sinclair ZX 81 BASIC Course	@ £17.50	
Basic Course Programs on Cassette	@ £ 2.50	
Not Only 30 Programs/Sinclair ZX 81:1K	@ £ 6.95	
Machine Language Programming Made Simple	@ £ 8.95	
Understanding Your ZX 81 ROM	@ £ 8.95	
Postage and Packing		£ 0.80
Remittance enclosed		£



NOW YOU CAN BUY THE BEST HOME COMPUTER IN THE WORLD.

Commodore's brilliant new VIC20 home computer is at your dealer's today. Now you can get your hands on the world's best home computer, we're sure you won't be disappointed.

Here's a brief reminder of VIC's many features:

A typewriter keyboard with graphics.

5K Memory–expandable to 29.5K user RAM.

8 programmable function keys.

High resolution graphics.

Music in three voices and three octaves, as well as sound effects.

Eight border colours and sixteen screen colours.

All in a superbly-designed, compact unit with built-in facilities for expansion.

Now you can take a look for yourself. And even take VIC home tonight.

commodore COMPUTER

For more information on the VIC 20, telephone or write to: Commodore Information Centre, Baker Street, High Wycombe, Bucks, or Tel: Slough 79292.

Notting Dale Technology Centre

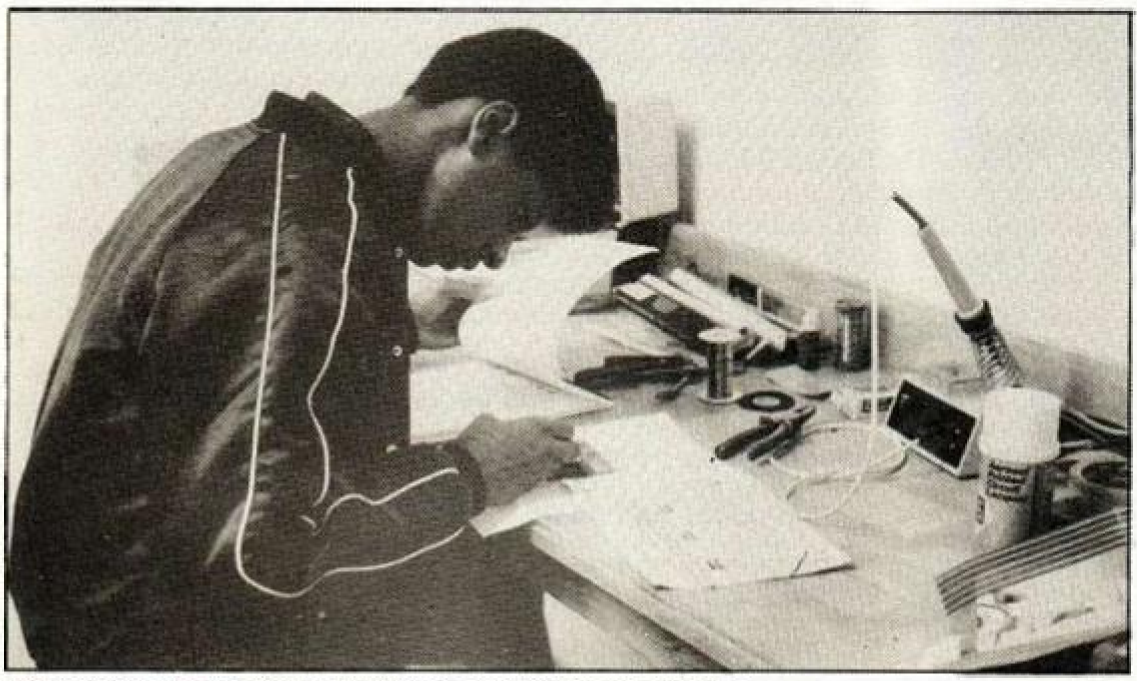
An inner-city success story can be seen in action in a West London workshop where up to 30 youngsters at a time are using the facilities of the Centre to build themselves the foundations of a career. David Pollard reports on this Government-funded project

FOR ONE UNACCUSTOMED to believing that State-financed schemes ever do much good, a visit to London's Notting Dale Technology Centre proves that this time a Government plan is going to work.

Some 18 months ago, Chris Webb established the Technology Centre at Notting Dale in the back streets of Notting Hill Gate. Many of the slums have been replaced in the last few years but the area as a whole remains poor. It is hardly the place where you would expect to find a computer centre, sited opposite the Free Republic of Frestonia — a group of houses occupied by squatters a few years ago.

Four years ago, Webb had been working on community projects with 16- and 17-year-olds who were mainly unqualified. No status, no jobs, no prospects, poorly housed. He had foreseen the bleak prospect of six or seven million unemployed by the late eighties. Reading research papers, seeing the engineering industry contract and computing grow, he decided to "dance with the silicon devil".

With the assistance of interested staff at Imperial College, the Institute of Education and the Harrow Trust — a local charitable organisation — he started putting resources together. £750,000 was raised. A disused bakery was bought and converted, ready for the computers and peripherals.



Work at the centre includes development of aids for the disabled

Eventually, the Manpower Services Commission (MSC), succumbed to such a strong assault. Now it will be funding 30 similar centres throughout the U.K.

For those contemplating similar projects, the moral is clear: amass what you can from all the resources available; group together with others of like mind for strength, and start your computer or electronics centre. When and if it works, officialdom may well offer some tardy assistance; by then you can, to an extent, dictate your terms.

The notion that Webb had was a complex mixture: there exists a mis-match of jobs and skills; job creation is best locally controlled; unemployed youngsters have plenty of nous and little schooling; education is most effective when motivated by a need to know; integration of real practical work experience is a key feature of adult education.

As an act of faith, on a first-come, first-served basis, and with self-motivation as the main criterion for acceptance, 30 youngsters were taken on for a year as a Youth Opportunities Programme (YOP) project. Word of mouth seems to have been the main advertisement. Those who found out about and visited the Centre were presumably the more strongly motivated.

With a teaching staff of six, 15 Pets, several Aim-65 boards, sundry other computers, test gear, etc., their learning was in four main areas: the modern electronic office; electronics, with an emphasis on the digital; programming, mainly Basic and Logo with some machine code; and prototype development which was directly practical, starting with no previous work experience.

The success rate has been astounding. Some 65 percent are now employed. No wonder the MSC was so keen to help; it cannot achieve this rate with university graduates, let alone poorly-educated youngsters.

The regime, if such it can be called, is essentially informal. Though lateness or non-attendance leads to a docking of the £23.50 YOP wage, there is no strict timetable or curriculum. If someone should wish to play Space Invaders for two months to remove all trace of it from his system — it rarely takes longer — then he will probably be left alone to do it. Sooner or later, with this laissez-faire approach, a working understanding is reached. Only by giving the responsibility of action can responsibility be developed.

The youngsters have two definite assets — they know when they do not know something and they have commonsense. Given a toolkit — soldering iron and pair of pliers — and a few components, they can start making things. When they realise they do not know something, then it is time for theory. For example, they might ask for a seminar on power amplifiers — to make an electric guitar sound better — or on analogue interfaces to input in a particular way to a computer.

One of the weaker areas is programming. The greater proportion of software uses mathematical modelling. How, then, do you teach someone with a totally non-mathematical background? New languages are needed as are means of incorporating basic mathematics into the brief 12 months.

The trainees are very good at problem solving and this shows, when they are designing. Having gathered components, discovered the necessary theory, in the end they usually find a very elegant final solution.

Where are the jobs at the end of this? Some youngsters have gone into apprenticeship to further their careers. Sound-recording studios, musical-equipment manufacturers and Space Invader emporia have taken on trainees. Further education, working for the Prestel service, and salesmanship are courses taken by others.

After a year's training, they will have acquired some of their tutors' experience, and the Centre clearly enhances credibility with local employers. By working with high tech-

(continued on next page)



(continued from previous page)

nology, in an area of high potential, their self-respect and self-confidence is also improved.

A few jobs have been created at the Centre. Integrated into the scheme of things at the Dale are two businesses. As with any form of further education, real products are developed and manufactured alongside the educational process.

Simon Browning runs a small firm developing aids for disabled people. With a start-up capital of £40,000, they have a year in which to become established. If all goes well, similar firms could be attached to the 30 centres throughout the country. Such a concept is, indeed, a powerful one.

Not only does the design and development of electronics-based aids provide teaching material and direct experience; the manufacture and servicing will provide worthwhile jobs and much-needed equipment for the one in 10 of the U.K.'s population who are disabled.

Local support can be provided at the Centre, backed by the teaching staff and engineering associates. Networking between centres means experience and design work will be shared, so cutting development costs.

Need-based — as opposed to consumer-orientated — manufacture is not the easiest of areas in which to start a business. There is little money available. The demand is there and there are many disabilities which can be lessened through the appropriate use of high technology.



If you want to know what devices they will be making, imagine yourself paralysed, or blind, or deaf. Now think how you could use your computer to help. There is plenty of scope.

If you spawn an original idea, forget it for a few weeks. Then if it still seems good, write it down and send it to the Centre.

Richard Hillier is a well-qualified electronics engineer who has set up Countermeasures Ltd alongside the Centre. Having developed high value-added .products — specialised data loggers, a polyphonic synthesiser, an EPROM blower, among others — the deal is a straightforward one of a 50/50 split of profits in return for work space. It looks like a stable and worthwhile interdependence.

Of course there have been problems; there have been breakdowns and not everyone has completed the course. As a whole, it's a definite success. The fact that the Department of Industry's Education Department will be investing £6 million to £7 million in creating similar centres throughout the U.K. over the next year is a measure of that success.

Chris Webb and three colleagues will be heading the DoI consultancy set up to handle this development, which means that it stands a better-than-average chance of success. They will provide technical and educational advice at an adult and community level as the new units are started in tandem.

The Information Technology Centres (ITECs), as the Press has nominated them, will initially be situated in the depressed inner-city areas and will be modelled on the Notting Dale Centre, taking into account current needs, local and industrial resources.

The beauty of a loosely-coupled network is that it can be tailored to suit and adjust to changes in local and immediate requirements. Prestel will link the centres — information will be available on-line — giving an economy of scale, though autonomy can be a strong feature, integrating each unit into its individual environment. ■

Computer Club is here to encourage you to start your own local computer club or, if one already exists, to join it and become involved. Each month we will devote the page to new ideas from local clubs. We would like to hear of anything which has made a club a success, or of any projects or programs you are developing.

INNOVATIVE
TRS 80-GENIE SOFTWARE
from the professionals



.... a new, simple to use, moderately priced word processor

The introduction of a brand new word processor is a major event and AJEDIT is without doubt a major program. There are, however, quite a few Word Processors around and most of them are extremely good ones - why, therefore, another? The question is even more pertinent when it is known that we specifically commissioned the writing of it from an author of the status of Denville Longhurst of Enhanced Basic fame. The answer is that user feedback shows that a large number of customers do not need or want word processor programs which require a quantity of training before use. Scripsit, for instance, is an excellent program, but is complex to use; it even comes with a training course on tape. If one operator is dedicated to using the word processor then it makes sense to have her trained, and the more complex the program (so long as the complexity is accompanied by more and bigger functions) the better.

AJEDIT has been written for the user who needs a word processor intermittently, say three or four times a week. Its prime design criteria was ease of use - and just as importantly - ease of recollection of its commands. Take, for instance, the text editing commands - they are as close to the Basic Edit commands as possible, so that the user will remember them: To insert type 1, to delete D, to take out three letters type 3D and so on.

Furthermore, AJEDIT has benefited from being written after a number of other word processors. The deficiencies in its predecessors are corrected in AJEDIT. For instance, any control characters can be outputted so that full advantage can be taken of the features of the particular printer being used. Disk directory access is available from within AJEDIT as is the killing of files on the disk. The FREE command and a number of other DOS commands can be carried out from within the program with a return to AJEDIT - with its text intact.

AJEDIT contains close to one hundred commands covering most word processor requirements. Dedicated printer commands for the Epson MX series and the Centronics 737 are included - again for ease of use of these two popular printers.

AJEDIT needs 48K and one disk minimum and is suitable for the TRS-80 Models I and III and the Video Genie Models I and II.

AJEDIT **£49.95**
Inclusive of V.A.T. and P. & P.

MOLIMERX LTD
A J HARDING (MOLIMERX)

1 BUCKHURST ROAD, TOWN HALL SQUARE, BEXHILL-ON-SEA, EAST SUSSEX.

TEL: [0424] 220391/223636 **TELEX 86736 SOTEX G**

TRS-80 & VIDEO GENIE SOFTWARE CATALOGUE £1.00 **[refundable]** plus 50p postage.

SURVEY

VIDEO GAMES

Selwyn Ward's look at the major video-game computers reveals them to be far more sophisticated beasts than their pedestrian forefathers. Their improved display and definition heighten the effects of the latest adventurous games.

RACING TO the finish on a Grand Prix circuit, manoeuvring round trees and moguls in a downhill ski run, destroying hordes of mal-intentioned Martians, and averting a nuclear holocaust, are some of the more soothing armchair experiences offered by the latest generation of plug-in TV games.

TV games have advanced a long way since the beep-beep of ping-pong tennis and soccer games filled Christmas stockings not so many years ago. Now, games are altogether more sophisticated, considerably more expensive, and usually known by the more grandiose title of "video computer games".

With Christmas rapidly approaching, we look at four video-computer game systems: Atari VCS, Philips G-7000, Mattel Intellivision, and the Interton VC-4000. The Interton is similar to the Acetronic MPU 1000, Radofin 1292/1293, Prinztronic VC-6000, Teleng, Rowtron and Database computer-game systems.

All the video-game computers plug into the aerial socket of a conventional television — preferably colour, as they all generate colour graphics. All are equipped with game controls and a mains transformer. In the case of the Mattel Intellivision, the transformer is in-built.

The video-game computers are described as "programmable", although this does not generally mean that they can be programmed by the user. It refers to the fact that video computers can accept plug-in cartridges which allows you to have a continually-expanding library of games. Inevitably, cartridges produced for use with one video-game computer system cannot be used on another.

The Atari VCS has been on the scene for longer than its rivals and with a range of around 40 cartridges, it has by far the largest selection of games available. Cartridges vary in sophistication from relatively simple bat-and-ball games to complex animations such as Superman where action is limited not solely to the dimensions of the TV screen.

The player — controlling an animated Superman figure — has to roam through a series of inter-connected displays, flying out of the left-hand side of one display into the right-hand side of the next, to capture and jail a band of desperadoes, find and rebuild the hidden sections of Metropolis Bridge, and change back to meek mild-mannered news reporter Clark Kent to file his story at the *Daily Planet*. All of this while dodging Kryptonite meteors which rob Superman of his powers — and for which the only cure is to find and kiss Lois Lane.

Cartridges are also available containing excellent chess and checkers programs — I found that the Atari chess cartridge could usually better my dedicated-function Boris chess computer — as well as backgammon and Othello, which is also known as Reversi. A Basic Programming cartridge is also available, although it is extremely limited in scope.

The Atari is best known for its versions of popular arcade games. Most video computer systems now include a Space Invaders cartridge in their range, but the Atari cartridge most closely reproduces the arcade game which has spoiled so many pubs throughout the U.K.

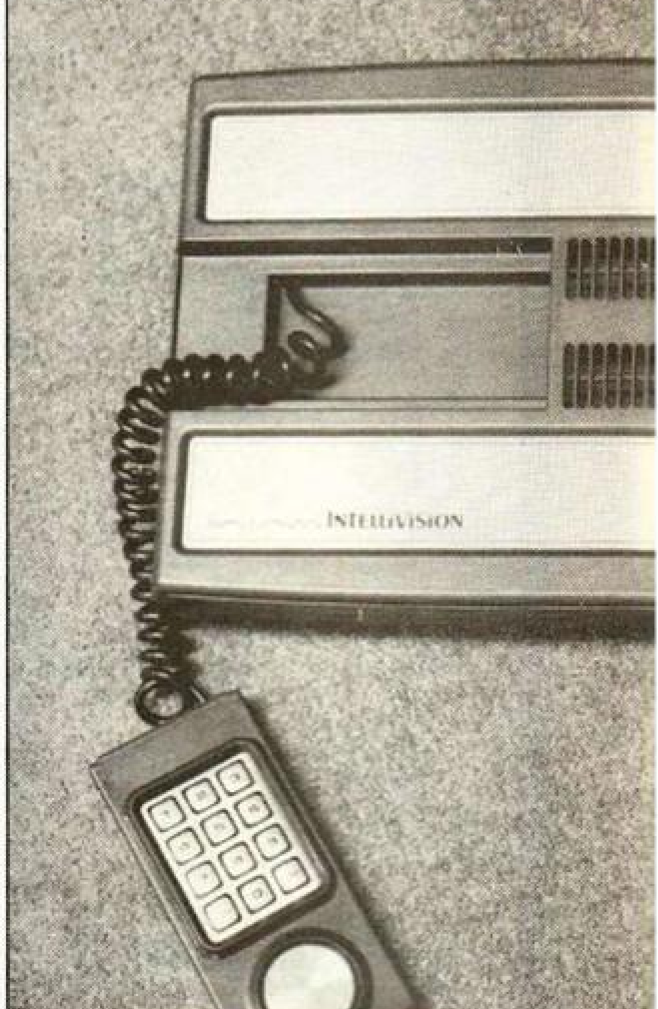
New Atari cartridges include Asteroids and Missile Command; again, both are based on arcade games and both have very fast action, indeed. New cartridges fully compatible with the Atari are also now being produced by an independent company, Activision. These already include some three-dimensional sports games which compete directly with the generally more sophisticated sports-game cartridges available for the Mattel Intellivision.

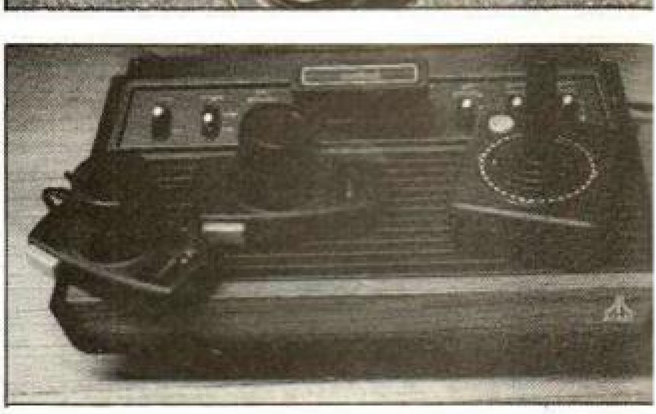
Separate joysticks

Unlike the other video-computer game systems which have multi-function game controls, the Atari uses separate joysticks and paddles, which means users have the task of plugging and unplugging controllers when changing between some of the cartridges. It also means that to use some cartridges, additional controllers must be bought. In fact the controllers seem to be the weakest feature of the Atari.

The joysticks in particular are prone to jamming or breakdown. On the other hand, the Atari is the only video-game computer which — at least with certain cartridges — allows four players to compete simultaneously, provided you buy an additional pair of paddle controls.

The Atari VCS, which can be found for slightly less than £100, is complete with a Combat cartridge which comprises a variety of tank and air-battle games. Additional cartridges vary considerably in price from around £16 to £35. The newer cartridges tend to be in the £23 to £29 price range, although the Activision range costs around £16.











Above: the Interton VC-4000 is based on the same clip as the Acetronic MPU 1000, Radofin 1292/1293, Prinztronic VC-6000, Teleng, Rowtron and Database systems.

Top left: the Intellivision from Mattel.

Far left: there are 40 cartridges in the Atari VCS games library.

Below: Philips' G-7000 video-game machine.



The Philips G-7000 video game computer is an attractively-styled piece of equipment with a full alpha-numeric, touch-sensitive keyboard — although most of the Philips game cartridges use only the system's joystick controllers.

The release of cartridges for the Philips G-7000 started slowly, although there are now about 25 in the range. As with the other video-computer systems, the range is varied and offers sports games, shooting games and maze-chase games. The number of game variations on each cartridge is mostly very limited, although some of the early Philips games cartridges included some curious combinations: one cartridge contained the odd mixture of anagram and car-race games; another contained a lunar-landing game, a memory game and a Mastermind code-breaker game.

Graphics on the Philips G-7000 are relatively crude. Often objects which are displayed appear as seemingly abstract shapes. There are occasional compensations, however — particularly entertaining is the animation of the swaggering gunslingers in Gunfighter, who closely resemble a pair of saddle-sore John Waynes.

Some of the Philips cartridges are cleverly conceived and are peculiar to the Philips system. Stone Sling, for example, is a two-player game where each player controls a medieval soldier and stone catapault behind a towering fortress.

Players fire their catapults at each other to eliminate their opponent's fortress or hit the opponent's catapult. The fortresses crumble a little more each time they are hit until one is completely destroyed. Then the losing soldier waves a white flag to surrender. One of the newer cartridges, Basket Game, is also unusual in that it is, in effect, a two-player version of the usually solitaire pinball game.

Limited scope

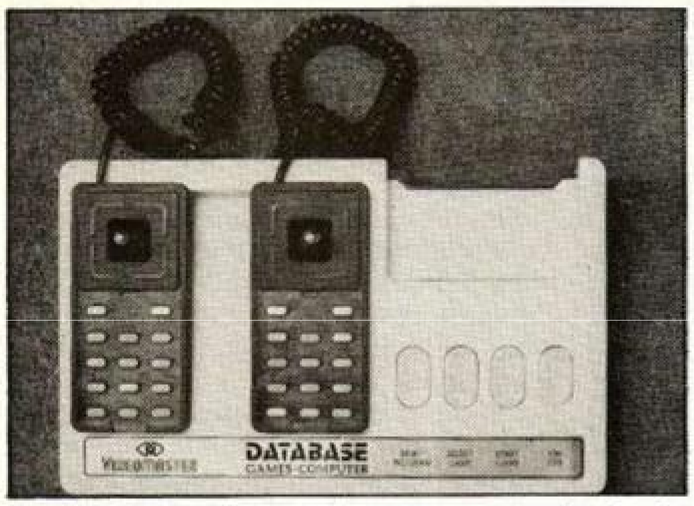
The Philips G-7000 retails at about £95 and all cartridges are the same price at about £15. An assembler programming cartridge is available which compares favourably with the Atari Basic Programming cartridge, but which is still limited in scope — not least because there is no means of storing any program which a user does manage to write.

The most expensive of the video game computers and in many respects the most sophisticated, the Mattel Intellivision system combines superbly-detailed graphics with relatively realistic sound effects. A range of around 20 cartridges is currently available.

The Mattel controls differ considerably from those of rival systems. Each control has both a touch-sensitive keypad and direction disc. With every games cartridge you are provided with a pair of overlays which slide on top of the game-controller keypads and indicate the function of each pressure-sensitive area. The direction disc corresponds in function to a joystick.

I found the overlays a very useful innovation, particularly given the number of functions used in some of the more complex games. I would, however, have strongly preferred a good, old-fashioned joystick to the touch-sensitive disc as a means of control.

(continued on next page)





Left: the Database games computer and above, two of the latest cartridges for the Atari VCS.

(continued on previous page)

Many of the Mattel cartridges feature a very acceptable simulation of three-dimensional graphics. The tanks manoeuvring in Armour Battle are no mere missle-spouting blobs, but detailed figures — although despite this, and the large variety of terrain displays which can be randomly generated, the game has if anything less potential than the more conventional tank-battle games available with the other systems.

More complex are the battle games which feature both strategic- and tactical-level combat. In Space Battle, for example, play begins with a radar display showing five fleets of alien space ships moving in from different directions towards the player's mother ship. The player has three squadrons, each of three fighter ships, to launch against the approaching enemy.

The radar display is used to deploy forces, but where a squadron intercepts an alien fleet, the player may switch to a tactical display where he can view the action as if from the cockpit of one of the fighter ships. The object at this stage is to shoot down the alien ships while avoiding their laser fire. Players can return at any time to the strategic radar display to check on progress of the squadrons and of invading fleets.

However, where the Mattel system moves into its own is with its range of elaborate sports games. These involve fine detailed graphic displays showing complete animated teams, playing full-length games, and even featuring crowd noises. The team-sports games actually seem more complex than they really are, as players each control only one figure at a time while the computer animates all the remaining figures, but they are certainly entertaining to play and watch.

In Hockey, which is actually an ice hockey game, the cartridge even provides for fouls and player figures being sent off to the penalty box. In Skiing, among the most enjoyable of the Mattel cartridges, players take turns to race against the clock and/or each other on a variety of downhill or slalom runs, dodging trees and jumping moguls, manoeuvring the skier figure by altering the angle of his descent down the mountain. A Soccer sports-game cartridge is supplied with the Mattel Intellivision video-game computer.

The Mattel retails at around £200, which is perilously close to the price of some genuinely programmable, and more versatile, home computers.

There is, however, no programming cartridge available for the Mattel, although plans have long been announced for the introduction of an add-on keyboard to convert the Mattel Intellivision into a home computer. There is still no news of when the keyboard will be available in the U.K. nor of how much it will cost. Meanwhile, Mattel cartridges sell for around £19.

Typical of a number of video computer games, each with similar range of cartridges, there are currently around 25 cartridges available for the Interton VC-4000.

The Interton uses multi-purpose controllers, comprising a joystick attached to a keypad over which overlays can be fitted. I found these controllers easier and more convenient in use than those of the other video computer systems.

The cartridges themselves are varied but generally unremarkable. Some of the combinations are very good value — Car Racing includes games where the object is to avoid crashing into on-coming vehicles, games involving a race around a circuit, and night-driver games where the object is to stay on a sharply winding road. In the Atari system, where versions of the same games can be found, these are spread over three separate cartridges.

Nevertheless, the graphics and sound effects generated by the Interton system are extremely crude by comparison with the Atari, Mattel and even Philips systems. For some inexplicable reason, tanks or planes blown up in Tank/Air Battle do not explode, they enlarge to several times their original size.

Also in this cartridge, the planes in air-battle games have the irritating habit of freezing in mid-air when they reach the edge of the screen rather than flying around the other side of the screen as is the case with similar cartridges for rival systems.

The Interton VC-4000 retails for around £95, including a Space Invaders cartridge. Additional cartridges are around £15. Some of the similar video-game computers can be found from around £65, although usually the cartridge supplied is only a bat-and-ball game cartridge.

Video-computer games are likely to prove more than just a five-minute wonder in most households. With the new cartridges for all of the systems being released all the time, it is likely that your bank balance will run out long before your family's enthusiasm.

It is a sure bet that Santa will be delivering many video-computer games this Christmas — but do not be surprised if he's still elbowing his way through the kids to the front of the TV well into the new year. ■

HOME COMPUTER RETAILERS

BEDS
Computopia, Leighton Buzzard (376600)
Comserve, Bedford (216749)
Electron Systems (Sandy) Ltd, Sandy (81195)
Texas Instruments Ltd, Supply Division, Bedford (67466)

CAMBRIDGESHIRE
Intelligent Artefacts
Orwell: 020 689.

DEVON
JAD Ltd, Plymouth (29038)
Peter Scott (Exeter) Ltd, Exeter (9076842)

ESSEX
Maplin Electronic Suppliers,
Westcliff-on-Sea (554000)

HANTS
The Maths Box, Southampton (22958)
RDS, Portsmouth (812478)

HEREFORDSHIRE
Acoutape Sound, Bromyard (3280)

HERTS
Computer Centre (Watford) Ltd, Watford (40601)
Computer Plus, Watford (020449)

HUMBERSIDE
Radius Computers Ltd, Hull (227181)
Peter Tutty & Co, Hull (41458)

LEICS
Boots Company Ltd, Leicester (21641)

LONDON
Adda Computers Ltd, W8 (579 5845)
Eurocalc Ltd, EC2 (727 4555)
Landau Electronics Ltd, W1 (642 5986)
McDonald Stores, W1 (636 2877)
Mountaindene Ltd, NW11 (455 9823)
Selfridges, W1 (629 1234)
Sumlock Bondain Ltd, EC1 (250 0505)

NORFOLK
Anglia Computers, Norwich (29652)

NORTHANTS
Computer Contact, Rushden (55673)
Computer Supermarket, Corby (61587)

NOTTS
Bestmoor Ltd, Nottingham (41551)

OXON
Science Studio, Oxford (54022)

S. GLAM
Computer Business Systems Ltd,
Ely, Cardiff (562255)

S. YORKS
Datron Interform, Sheffield (58549)

W. MIDLANDS
Taylor Wilson Systems Ltd,
Dorridge, Solihull (79404)

W. SUSSEX
Gamer, Brighton (698424)

W. YORKS
Ackroyd Typewriters Ltd, Bradford (31835)
Bits and PCs, Wetherby (63744)

SCOTLAND
Esco, Glasgow (427-5497)
Robox Ltd, Glasgow (221-5401)

IRELAND
Texas Instruments Ltd,
Supply Division, Dublin (609222)

RUMBELOWS STORES STOCKING HOME COMPUTER

BEDS
Bedford (59339)
Dunstable (602618)
Arndale Centre, Luton (414480)

BUCKS
Bletchley (73505)
Central Milton Keynes (662120)

HERTS
Boreham Wood (953-1744)
Hatfield (64211)
Hemel Hempstead (53225)
Hitchin (59203)
Hoddesdon (66486)
Letchworth (71231)
Potters Bar (58804)
St Albans (532290)
Stevenage (54303)
Waltham Cross (22100)
Ware (2316)
Welwyn Garden City (27125)
Enfield, Middx. (363 7384)



The TI-99/4A The Home Computer worthy of the name.

Even if you're new to computers, you'll be using the TI-99/4A within minutes of plugging it into any standard TV set. Because the TI-99/4A is a true computer for the home. Immediately accessible to the whole family. All for around £299.

Just snap in one of our wide selection of Solid State Software* Command Modules, touch a few keys, and you're ready to go. The 40 modules can sharpen your children's maths, teach you to win at chess or even help you with household financial decisions. And much more besides. In all, over 400 programs are available.

All enhanced by full music capability and 16-colour graphics.

Another development which sets the TI-99/4A apart from the rest, is our optional Solid State Speech* Synthesiser. Actually reproducing the human voice. With our new Emulator Command Module* its vocabulary is unlimited.

For data input/output you can use an ordinary audio cassette recorder. And a full-size professional keyboard makes it easy to use.

The TI-99/4A comes with TI BASIC built in. Ideal for when you want to learn programming – and to get you started there's our "Beginner's Basic" course, free with each machine – yet powerful enough for even the most experienced programmer.

To help you get the most from the TI-99/4A you can join the independent users' club. And there's a special magazine ("99'er") available through dealers, or on subscription.

So, if you're looking for a home computer, you can't afford to miss the TI-99/4A for versatility, power and value for money.

- * 16K RAM. Expandable to 48K.
- * 26K ROM including 14K BASIC.
- * Command Modules add up to 36K ROM.
- * 13-digit floating point.
- * Other languages soon – logo, UCSD PASCAL, TMS 9900 Assembler.
- * Options – Speech Synthesiser, Thermal Printer, RS-232 Peripheral Adaptor, Disk Memory System.

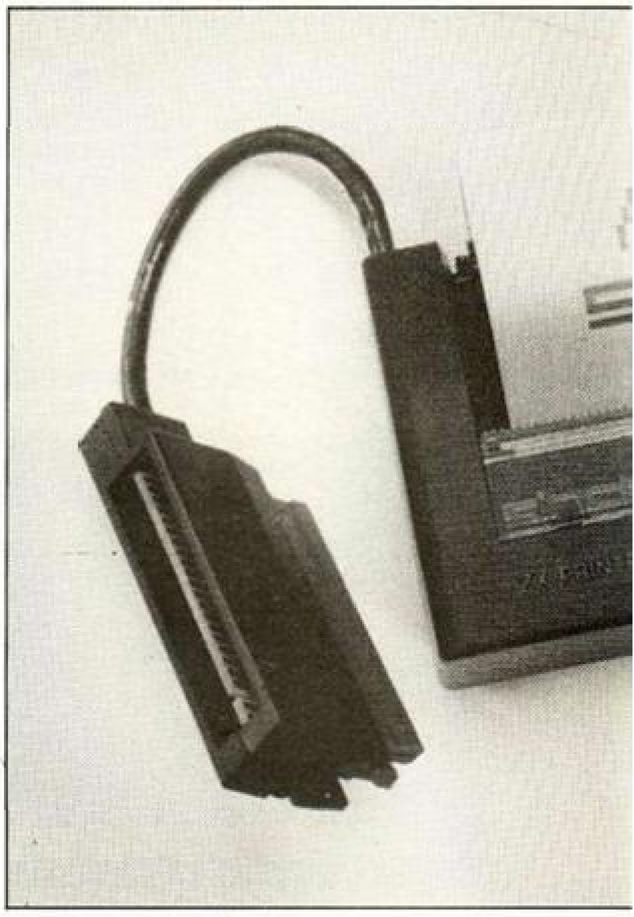
To find your local dealer check the list on the left, or for more information write to Texas Instruments Ltd., (MS.24), Manton Lane, Bedford NK41 7PA.

TEXAS INSTRUMENTS
LIMITED

*Trademarks of Texas Instruments.

REVIEW

ZX PRINTER



The new £50 printer from Sinclair means hard copy at a soft price. Eric Deeson assesses the device.

PERHAPS £50 seems a considerable sum to pay for an add-on to a £50 computer, but the ZX printer is a miracle at this price. Superbly designed, the cigarette-box-sized machine is inexpensive to run and gives impressive results. If you bear in mind that it speedily reproduces all the Sinclair graphics, you soon realise that it would be remarkable at several times the price.

Mechanically, the system is a neat application of a standard approach. A spark-emitting stylus scans the aluminium-coated paper, burning away the metal to create the black points which form the "printed" character. The line of sparks is a pleasure to watch. There is a very slight smell of burning, and the residue of burnt paper soon lines the stylus tracks.

Print quality is good, although there is sometimes a slight distortion in the verticals. Occasionally, the black smears on to the white and makes inverse characters rather hard to read. Even normal characters can be confused more readily than on screen — "I" may be mistaken for 1 and the numerals 3, 6, 5 and 8 are sometimes indistinct. This makes reading ZX printed listings rather tiring — especially with graphics and machine-code routines. Generally, print quality is perfectly adequate — photocopies made from the listings are good.

Technical Specification

Power Pack
Size: 10 by 7 by 6.5cm.
Weight: 625gm.
Output: 9V. 1.2A. Unregulated.
Cost: Included in price of printer.

Printer
Size: 14 by 9 by 5.5cm.
Weight: 400gm.
Power: From the computer.
Paper: Electrostatic.
Rolls — 19.8 by 10cm.
Five rolls £11.95 — one supplied.
Resolution: 256 by 256.
User graphics: Yes.
Expandability: Via connector to computer.
Cost: £49.95 including VAT, postage and packing.

The ZX printer is supplied in the standard-size Sinclair foam box, with a heavy new power supply, and a 16-page booklet with guarantee. The power supply replaces the one which fed the ZX-81 and is welcome because of its long leads. It becomes only slightly warm — the temperature of the computer gave me more concern.

The ZX-81 white-out problem is no more frequent than usual — even with the 16K RAM in position. However, one must now be even more careful during long programming sessions with the 16K. The printer plug fits into the RAM-pack position, and the RAM pack fits into the plug. The result is surprisingly stable, but must be less reliable than using the 16K connection alone.

The ZX printer is not silent — the corollary of its design. All the same, the noise is not excessive, and if there is vibration — as there must be — it does not disturb the nearby RAM.

The paper-roll holder is easy to attach and detach, to unload and load. When reinserting, however, push it home with an extra jog for luck to ensure that the clips are correctly located. The serrated tearer will presumably become blunt relatively quickly but works well as long as you hold the roll while tearing off the copy.

The printer's only hardware control is the paper-feed button. Very occasionally, this becomes stuck. In fact, I already bypass it in most cases by using Copy, Break. Otherwise, I use software control, as detailed in chapter 20 of the manual. While the printer is working, video synchronisation is lost — so, if you write a program mixing outputs to printer and display, do not put the instructions too close together; Pause 20 is enough.

The cursor homes, too, so you are obliged to reposition it using cursor control. The Copy command reproduces the contents of the screen-display area on the printer, including empty lines. It does not reproduce the message lines, so commands, report codes and input prompts do not appear. On the whole, that is acceptable, though I have already had occasion to wish it were otherwise. Break and Cont work with the printer as with the screen display, though Cont does not always give a perfect join. If-Then Copy is a useful technique.

Of course, you cannot use the printer for animated graphics games. However, repeated graphics patterns are really beautiful — the mix of striped black and silver is very striking. A few home users are bound to be tempted into creating designs for greetings cards.

The only standard printer feature missing is Echo — reproduce on the printer what is entered at the keyboard. This has a number of uses, not least allowing one to program away from the den without a bulky TV set. The obvious starting command is LPrint Inkey$, but that is not more use than Print Inkey$.

Technical

Stephen Adams approaches the ZX printer from the technical angle.

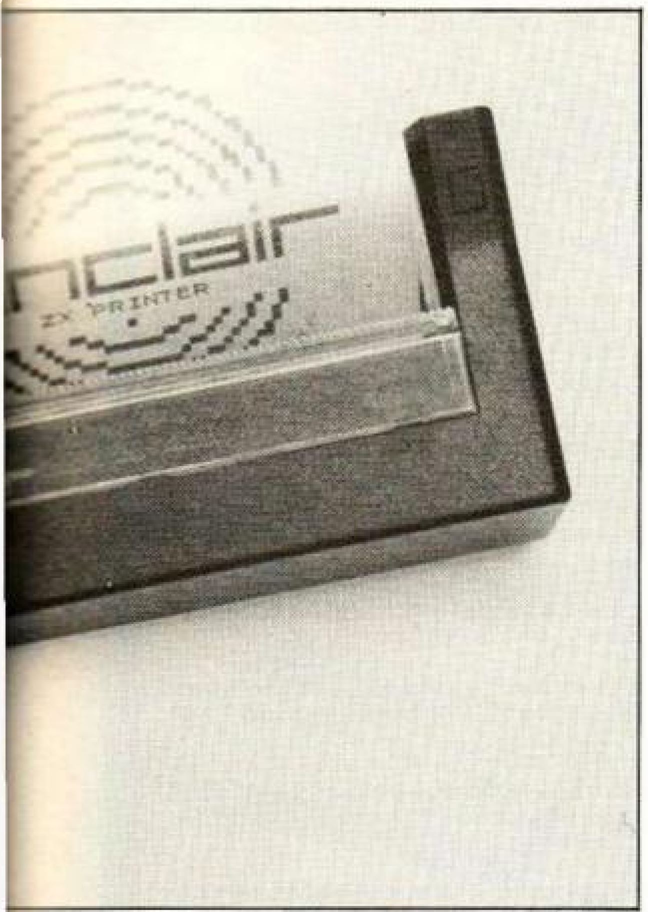
THE INSTRUCTIONS for most printers are at best sketchy, but those supplied with the ZX printer are most detailed and simple. The instructions for using the printer in a program are detailed on page 133 of the ZX-81 manual and are not included in the printer's documentation.

There is also a clear explanation, illustrated with drawings, on how to load the printer with paper which is the only complicated aspect of using the machine. Although Sinclair recommends its own aluminium-based, electrostatic paper, others can be used.

The feed button is the only control on the printer and when pushed, advances the paper a line at a time. Nothing can be printed while this button is depressed. When the printer has finished a 65ft. roll of paper, it needs to be cleaned. You can use a child's paintbrush to clear the burnt aluminium top paper from the recess under the paper carrier.

The printer is activated by addressing a port with address line A2 low. This is the usual form for Sinclair and, of course, reduces the number of input/output addresses available to the user. The official address for the printer is port FB which is both written to and read from. The printer also appears at other addresses so if you write machine code to access any one of these, you need to change it before you use the printer. A summary of the data bits and their uses are given in table 1.

The programs provided in the manual vary from a text justifier to a high-resolution plotter which enables the user to define every dot on



aspects

Table 1.

Data Bit	Use	Read	Write
D0	DOT STROBE	X	
D1	MOTOR SLOW		X
D2	MOTOR STOP		X
D6	$\overline{\text{PRINTER EXISTS}}$	X	
D7	HIGH VOLTAGE TO STYLUS		X
D7	STYLUS ON PAPER	X	

Note: Conditions active on binary 1, opposite on binary 0. All conditions are latched. On pressing feed button, D1/D7 are low and D2 high.

the printer. As this resolution is 256 by 256, there are 65,536 dots to define and as can be expected, that occupies 8K for the array alone. It means, however, that very accurate graphs and user-programmable graphics can output to the printer. Naturally, calculations concerned with these graphics slow down printing speed — in the Fast mode, the high-resolution plot program takes four minutes to complete.

All printing is done in Fast mode which means that there is no screen to watch while the printer is printing. A full screen can be printed by the command Copy and takes 12 seconds.

LPrint prints a line of text to the printer instead of to the screen and uses the same format as the Print command. The only command which cannot be used is At, but you can use Tab in its place as the LPrint prints only one line at a time. LList lists the program lines and can be set to any line in the program. LList will, however, only stop at the end of a program.

The printer is plugged into the back of the ZX-80 or ZX-81; its connector neither binds on the case or wobbles when a key is pressed. The connecting lead is only 3in. long, very stiff and must be placed on the right-hand side of the computer.

The 16K RAM pack plugs into the back of the connector if you have one. No problem was found in using the RAM pack since Sinclair Research has provided another power pack rated at 1.2A to replace the existing one.

The new power pack must be used — the old one rated at 700mA cannot cope. The new power pack has several design changes as it uses two power silicon diodes instead of the previous potted-bridge rectifier. It has also increased the size of the smoothing capacitors from 2,000μf to 32,000μf. It no longer plugs into the wall socket, but has a mains lead of 5ft. and no plug. This means you can position your equipment further from the power point, but you now have to go and buy a plug before using it. It would be better if Sinclair provided a plug so that the machine could be complete and ready to go as soon as you receive it.

Once the printer is connected, the power can be turned on. You will hear a short whirr from the printer as it lines up its starting point.

The printer is relatively quiet during normal operation, but when printing makes a noise like a clockwork toy, whirring and spluttering until printing is finished. The blue flashes under the transparent Perspex shield show the printer at work. Its stylus runs fixed to a continuous belt and burns away the aluminium.

When it has finished printing, the paper must be advanced — there is no spacing between pages — if you wish to tear the paper on the serrated edge provided. The print is very clear and the inverse graphics excellent. It can be very easily photocopied for a permanent record if you dislike the shiny surface.

A test was done on the paper to see if the print could be affected by heat as some electrostatic papers are. After three hours sitting above a gas fire, no deterioration could be noticed. The paper does, however, pick up grease from the fingers although it does not affect the clarity of the printing. Creases in the paper can crack the aluminium surface, but I noticed, no flaking of the surface.

The paper was also soaked in water and dried by a fire to test its extremes; the paper did not lose its white backing or the quality of the print. Altogether it seems satisfactorily indestructible. There are no gaps between the graphics so a continuous line can be created. Nor are there gaps between lines so you can print whole pictures — size of the characters is 2.5mm. square.

The paper is 10mm. or 4in. wide and only one 65ft. roll is provided with the printer. Replacement rolls are available from Sinclair Research at £11.95 for a five-roll pack.

Removing the top of the printer to look at its insides is not a practice I would recommend as traps lurk for the unwary which might wreck the machine.

The logic is contained in a single uncommitted logic array chip or ULA which has been specially programmed for Sinclair to control all the functions of the printer. It does several jobs:

- Puts out a strobe signal on Do.
- Controls motor on and off, plus the slowing of the motor near the end of the printed line.
- Checks that the stylus is resting on the metal strip on the left-hand side of the printer before printing.
- Decodes all addresses to the printer and latches them.

Apart from the ULA, there are various capacitors, resistors and a high-voltage transformer all mounted on a single-sided printed-circuit board. The motor and cable also terminate on this board. Under the board is a strobing disc which contains slots to tell the light-dependent resistor whether it is covering a black line or not. This makes sure that all 256 dots are printed in the same place each time and do not depend on the motor speed.



The motor drives a plastic worm gear to drive the paper up. A belt-driven system takes the stylus to where it should print each dot. This belt is carried between two pulley wheels whose ends are located in the top and bottom of the case. The belt is made of ridges which slot into grooves in the pulley wheels so there is no slipping.

On top of the belt the printing stylus is mounted — two pieces of wire are wound round a spigot on top of the belt. The wire is bent so that one end sticks outwards to write on the electrostatic paper and the other points inwards to connect with a metal track running inside the top of the case along the paper.

The stylus had to be made from special wire since it uses a high voltage to burn away the aluminium coating and must be able to tolerate the wearing away of the tip by these voltages.

The power for this machine is taken from the 9V line and this means that the voltage to the input of the regulator is lowered. It also means that the regulator runs cooler even though the printer does take some current from the +5V regulator.

Conclusions

- The printer is well made and robust and the paper-feed button is the only control on the printer.
- The printing quality is excellent and the paper is virtually indestructible.
- The paper can also be replaced by others although this is not recommended by Sinclair.
- Loading paper is simple and well illustrated.
- It is, on the whole a very efficient and above all inexpensive machine which should delight the user.
- The programmer is catered for under the technical section if he wishes to write his own printing routines.

INTERVIEW

RICHARD FOTHERGILL

Richard Fothergill heads the Microelectronics Education Programme which is backed by £9.5 million of State funds. Bill Bennett visited him at his North-East base.

THE PROGRAMME is run from a small semi-detached house on the campus of Newcastle Polytechnic, a short ride on the new Tyne and Wear Metro from the centre of the city. After we had met and I had been introduced to the staff, there was one burning question: "Why run such a seemingly important project from such a remote location"? The answer provided not only justification, but also a deep insight into the mind of the man who leads a team with a very important job.

"I was already working in Newcastle as a teacher trainer on this polytechnic site; I didn't really want to move anyway as I like it here. However, if microelectronics can do all that we are told it can, then it should present no problems. The one advantage of living in the microelectronic age is that one doesn't have to travel to work; the work can travel to you. Information transfer is readily available and we are using it".

What about the inevitable travelling that needs to be done in the course of your work? "Newcastle is fortunate in that it has excellent rail links: the Inter-City 125 trains mean that London is just over three hours away. Most of the travelling is done by the information team though.

"One advantage of our location is the peace and quiet — there is no busy traffic passing our windows all day. It means we are left to get on with our work here. The only disturbance is the occasional cheer from the football pitch". Cheviot House, the nerve-centre of the programme is located next to a sport field.

Having disposed of the questions regarding what now seems like a wise choice of location, I asked Fothergill to sketch the background to the DES Microcomputers in Schools programme. He at once pointed out that it was a microelectronics in schools and not just a microcomputers in schools programme, and that the scheme was to cover pupils from the age of five through to 13.

The sum of £9.5million index-linked to the 1979 value had been set aside for the programme by Neil Macfarlane, the then Parliamentary Under-Secretary at the DES, and the programme was to be run from March 1980 for five years. In November 1980 Richard Fothergill was appointed director.

"When I had assembled a team, we proposed a strategy. In effect, we began implementing our plan from February 1981, but it didn't begin officially until April. It was decided to concentrate on three main areas which, for the want of better titles, we have called: resource, teacher training and curriculum development, which is really a catch-all. We decided that because education is run in this country by local authorities, we realised the importance of involving them.

'We are on the verge of an explosion'

"The money we were given was not enough to make us significant to each individual local authority, so we divided the country into 14 regions; 12 in England, and one each in Northern Ireland and Wales. The Scottish run their own show from Glasgow. Both groups work in co-operation with each other.

"The resource part of the programme is designed to provide teachers with the support and information they need. It is vital that we can keep these teachers up to date with all the advances that are constantly taking place.

"This will be done by using regional information centres, exhibitions, as well as the more traditional access to books. There will be some machines and some software set aside for the purpose of demonstrations which should enable teachers to see the applications".

At the resource centre at Newcastle Polytechnic, in the middle of the floor, is an exhibition, mainly of posters. Around the edge of the room were all manner of teaching aids involving microelectronics, ranging from the Texas Instruments Speak and Spell game to Pet, Apple and Research Machines microcomputers, and even a word processor. To the side of the main room is a smaller one — this is where teachers are taught how to use the latest in new technology devices.

Teacher training is the second of the three points in the programme. "We train teachers to think of the computer as an instrument. We show them how it can revolutionise the office — indeed, part of the teacher-training programme is devoted to familiarising teachers with the electronic office.

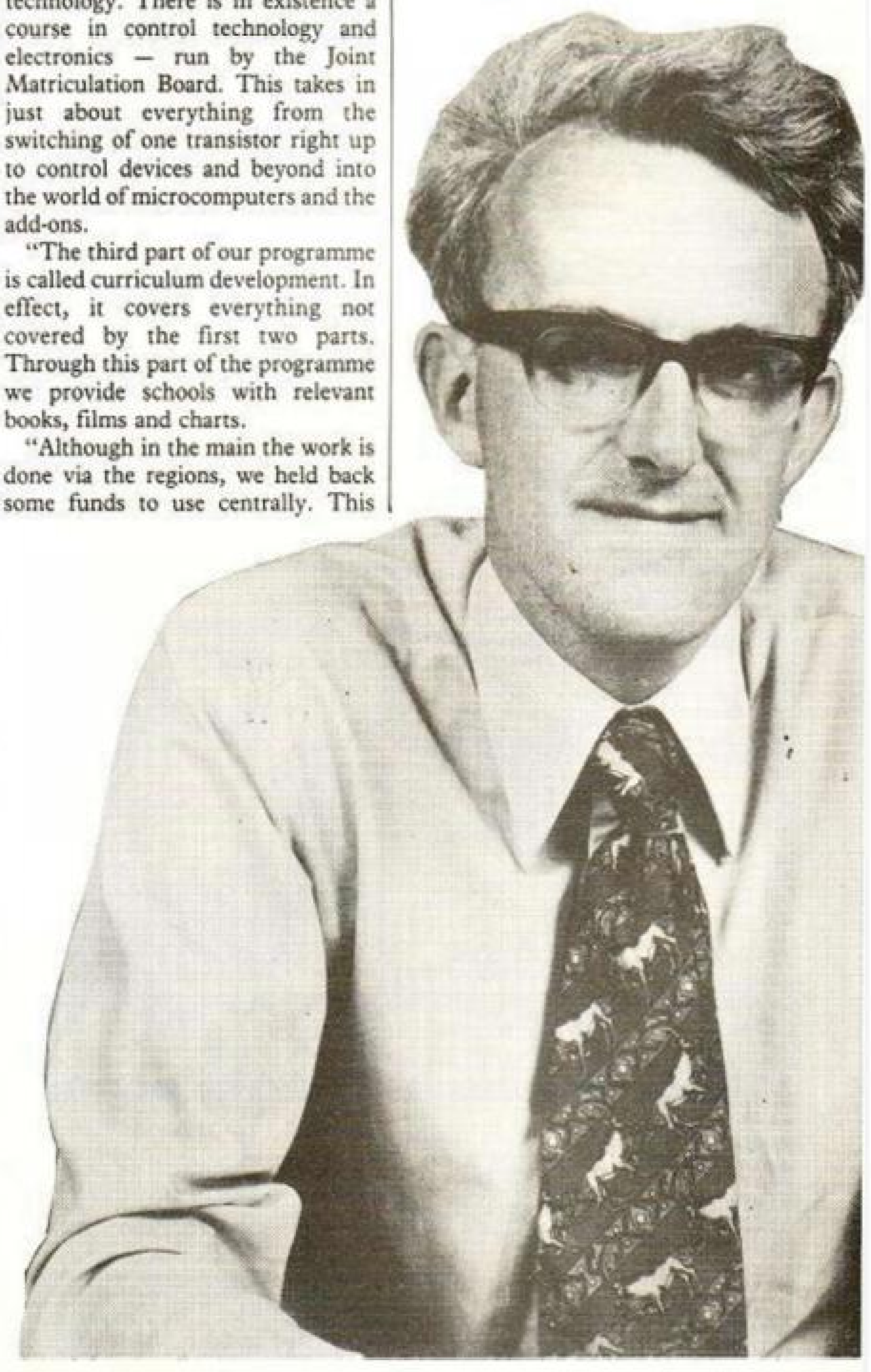
"We also train the teacher to use the computer for computer-based learning. I call it that because of the confusion between the existing terms such as CAL, CAI etc., which all mean separate things. Computer-based learning is my phrase to cover this.

"There is in fact another area in which we train teachers; that is in technology. There is in existence a course in control technology and electronics — run by the Joint Matriculation Board. This takes in just about everything from the switching of one transistor right up to control devices and beyond into the world of microcomputers and the add-ons.

"The third part of our programme is called curriculum development. In effect, it covers everything not covered by the first two parts. Through this part of the programme we provide schools with relevant books, films and charts.

"Although in the main the work is done via the regions, we held back some funds to use centrally. This money will be spent on what we call our national activities. The in-service training of teachers by the Open University is an example of this.

"One of the tasks facing us is the establishment of some standards, we will issue guide-lines on how to configure equipment, for example. This needs to be done because safety is an important factor in the classroom. We shall also advise on operating systems as well as languages and dialects. This will be done by a process of evolution — we will not lay down hard and fast standards, mainly because we couldn't make them stick. This means we shall have to evolve a set of agreed standards".



ON EDUCATION

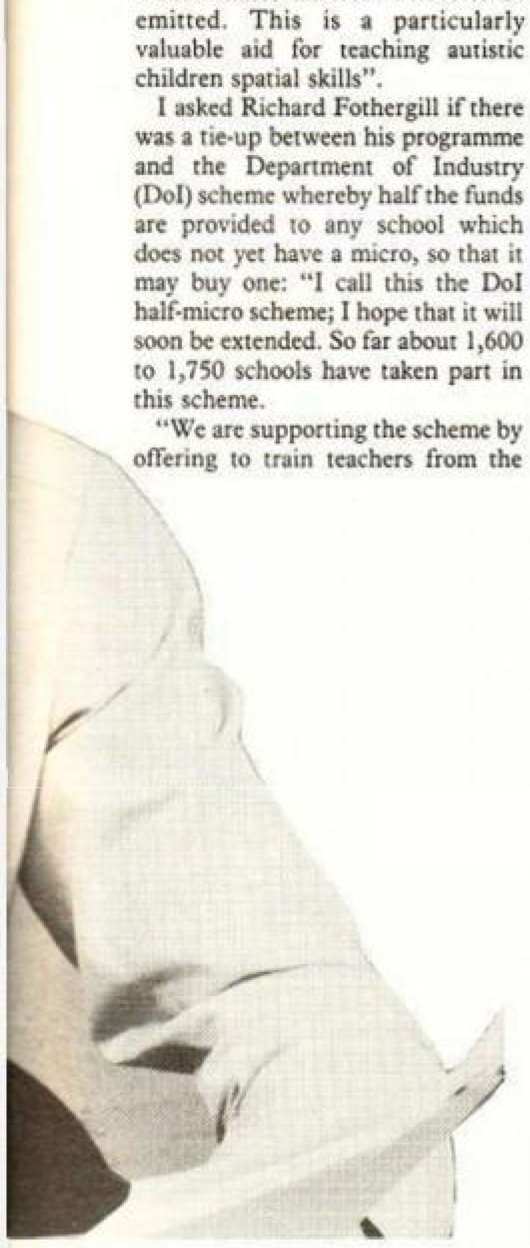
"Special education is an area which some of the centrally provided money is being spent. This will be used on making provision for handicapped and retarded children. There will be special software for those children with learning difficulties — for example, simple maths exercises.

"Special add-on devices can be used to help children express themselves. One such device is a box with coins struck on pads, the children can relate to this form of input easily. There is even a rubber for them to press if they wish to erase something.

"To the physically-handicapped child, the computer is more important as a communication aid. This is really a matter for the Department of Health and Social Security; we are more interested in education. However, one useful device is an upside-down Tupperware bowl. It has shapes on it: pressing these shapes in various orders causes different noises to be emitted. This is a particularly valuable aid for teaching autistic children spatial skills".

I asked Richard Fothergill if there was a tie-up between his programme and the Department of Industry (DoI) scheme whereby half the funds are provided to any school which does not yet have a micro, so that it may buy one: "I call this the DoI half-micro scheme; I hope that it will soon be extended. So far about 1,600 to 1,750 schools have taken part in this scheme.

"We are supporting the scheme by offering to train teachers from the schools involved. This means that two teachers from each school will take a four-day in-service training course to learn about their respective computer. We are also supplying training materials. However, the scheme does have its limitations.



"Of course, it is only correct that any Government-funded scheme should support British microcomputer manufacturers. But to simplify the administration of the scheme, there are only two packages available — a yes or no choice between two systems which represent the two ends of the price range. The choice of machines is restricted to the Research Machines 380-Z, which is commonly available and hence the best machine, or the new BBC micro, which looks as though it will give plenty of development possibilities.

"Under the circumstances, this is as good a selection as could be made. The Acorn Atom has a proven track record; the BBC machine will be in a form that is bound to be successful. The 380-Z is a machine which is well proven in schools and is a solid machine — it can take the treatment the schools will give it. The machines are the DoI's choice though, not ours".

So what kind of feedback has Fothergill received from the local

'Safety is important in the classroom'

education authorities (LEAs)? "Positive", was the immediate reply. "Many comment on the time scale of the project. For some it can never be fast enough; others have adjustment time problems. The LEAs have been supportive — they especially welcomed the teaching packs. On the whole everybody in the public sector is pressed for funds, so we are pleased that so many schools are prepared to spend money on microcomputers".

Many *Your Computer* readers will be interested to hear Richard Fothergill's ideas on selecting a home computer, especially with a view to education. First, I asked if he had any views on the trend towards home computing: "The future of the U.K. depends on the imagination of these people. A Sinclair connected to the television is more entertaining than the TV programmes on it". He points out that one of the aims of the



The MEP team from left to right: Mike Bostock, technology manager, Richard Fothergill, director, Helen Hinders, programme assistant, John Anderson, deputy director, Bob Coates, computing manager.

project is to stimulate and excite the next generation about microelectronics. He also says he would like to see an interface for the ZX microcomputers.

If a parent was considering purchasing a home computer for a child to learn programming, what should he or she look for: "First, a machine which can be attached to a TV, stable lettering, a good character display. A simple approach to Basic is important, as is a good manual — in fact there isn't much point without one. Good graphics would be a bonus. Simple dumping probably to cassette, but with the ability to expand to discs later, expansion to colour, communication, and viewdata should all be considered as well as the easy memory expansion.

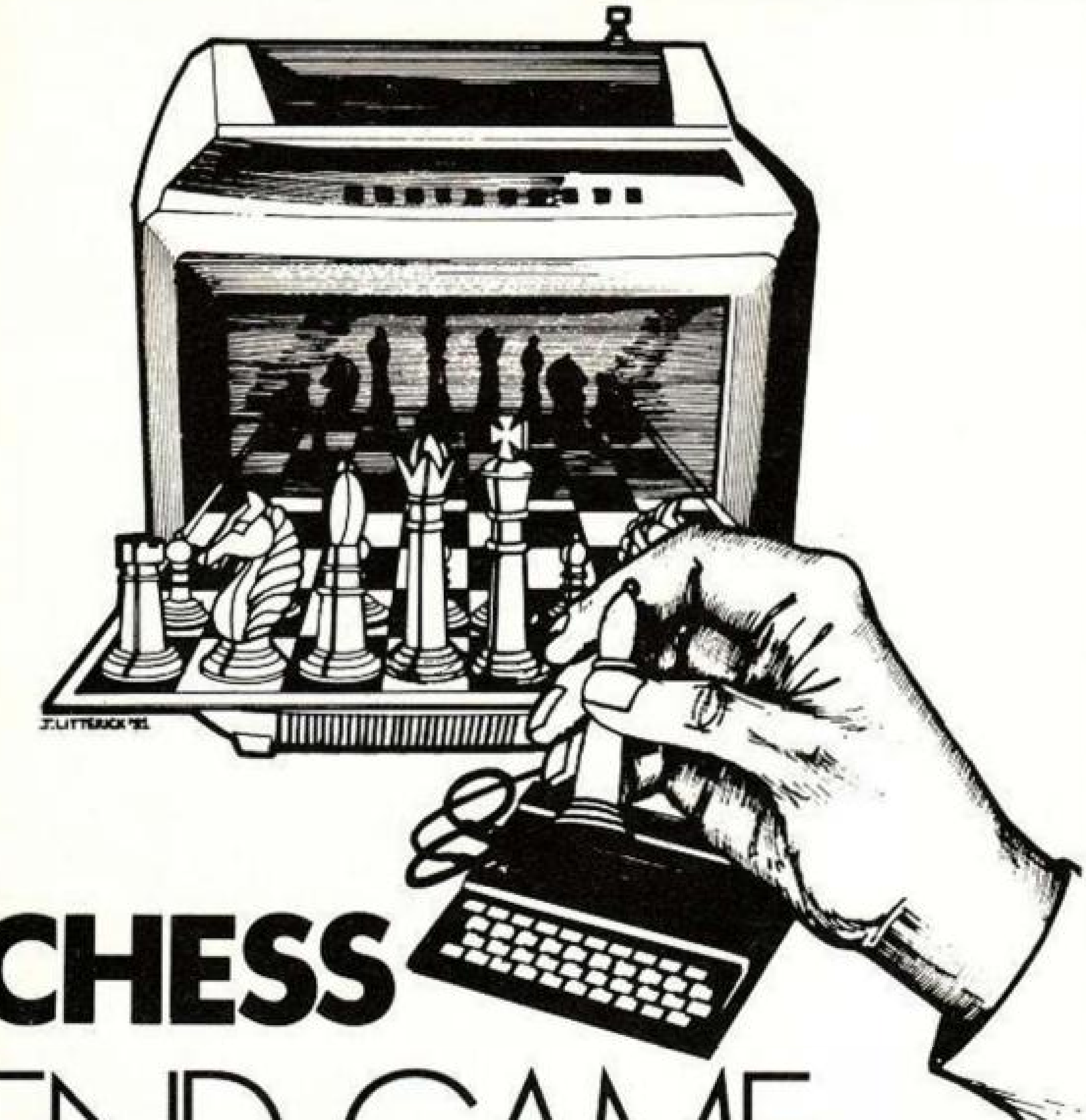
"If the children were under 12 then any good Basic machine would do. I think the RS-232 interface is preferable to the less popular Centronics type. With the possible exception of the Sinclair computer, a bare minimum of storage would be 8K. This would soon need updating to 16K, and as soon as the children want to use databases, as much memory as possible should be added".

Much has been said over the past year or so about the desirability of structured languages; what about this trend? "There is definitely a trend towards structured languages, but the initial hump of microcomputer work has been done. All future development must take account of this. Movement will be in this direction, as in the Basic chosen for the BBC micro, which is three-quarters Comal. Things like this cannot be done fast.

"The Microcomputers in Schools programme did not adopt a language as such — we were stuck with Basic, but let it evolve".

What about the less gifted children — is it possible that the introduction of microcomputers in schools will see them further and further behind? "Not at all, the early computer-aided-learning programs were in the main designed to help the less-than-average child. If anything, the microcomputer will help these children by giving them the plodding, patient tutoring that they need. In fact, it is a deliberate policy of the programme to ensure that the below-average child gets a chance.

"Education today is at a cross-roads, what we have now is a period of transition. For the first time the majority of schools have micros. In a year or two schools will have several more machines which will be scattered around in different departments — we are on the verge of an explosion". ■



CHESS END-GAME

Exotica such as the king ripple and the pawn-advance routine are two of the techniques John White has incorporated into his entertaining chess program in Basic, End-Game.

END-GAME has been written in Basic to complement the draughts program J-Checkers, published in the October issue of *Your Computer*. It exemplifies the method of move assessment known as iterative deepening.

I have chosen the end-game of chess as a model because it limits the number of pieces used and because the concept of mobility — essential in full games of chess — can, at a pinch, be ignored to keep the time taken for the game within manageable limits. I have eschewed fancy time- or memory-saving tricks for clarity.

Having tested this program, I have satisfied myself that it is not possible to write a satisfactory program for playing a chess end-game using a look-ahead of only two-ply. I hope this information will be of use to those contemplating writing their own chess programs.

End-Game does, however, play a fractionally more sensible chess end-game than many of the weaker chess computers available commercially, bearing in mind the fact that a compiled version would run in about two seconds. The interpreted Basic version presented here requires an average of two minutes a move.

The end-game of chess is hard for a human to play well, but very difficult indeed for a chess computer. A human can easily see at a glance what will happen six to seven moves ahead for both sides — grandmasters can see much, much more.

A chess computer will normally only analyse two or three moves ahead — four- to six-ply — although one or two of the most modern machines switch in extra routines for the end-game when sufficiently little material remains on the board. Under these circumstances up to five moves ahead — 10-ply — may be evaluated. Even so, the play is still weak by human standards. The classic problem is that shown in figure 1.

It is possible for a human to see at once that black's only sensible move is K-B6 — or B8 or B7. Anything else loses the pawn to white's attacking king. I shall avoid the problem of whether black can win even if he does save the pawn. Yet very few chess computers can see this solution, and most play pawn endings very badly, moving pieces almost at random.

Since the necessary deep search to play a good end-game is very time-consuming, I have tried in End-Game to produce an evaluation function which will play a recognisable end-game superior to that of most chess computers but using only a two-ply search. Essentially I have relied on the well-known maxim of "Push a passed pawn".

End-Game is written in Basic which imposes its own stunning restriction on what can be placed in the program: interpreted Basic runs some 200 times more slowly than the machine code used in chess computers and a complete game of chess is out of the question. Restricting the pieces to pawns and king only gives a respectable game with a clear objective: advancement of a pawn to the eighth rank.

The first player to do this has essentially won at chess, and has won End-Game outright. It may be noted that the powerful Sargon 2.5 and Morphy chess programs also adopt this policy in their end-game play, and will make any sacrifice to delay the arrival of an enemy pawn on the eighth rank.

End-Game uses a single subroutine to evaluate the position arising after each move — instead of evaluating the merit of each move itself, a strategy employed in other published games. The moves of each piece are generated by the program which assigns a score to the position arising from each move at the first level of search — one-ply.

The moves are then sorted, using a fast-sort routine which arranges the score in order of decreasing merit. The moves creating the scores are also rearranged, of course.

The program now calls itself — an example of recursion in Basic — to generate the responses to its sorted moves. It assumes that the opponent will be trying to maximise his score, and thus minimise the machine's score. So the best — lowest-scoring — opponent move is stored in location.

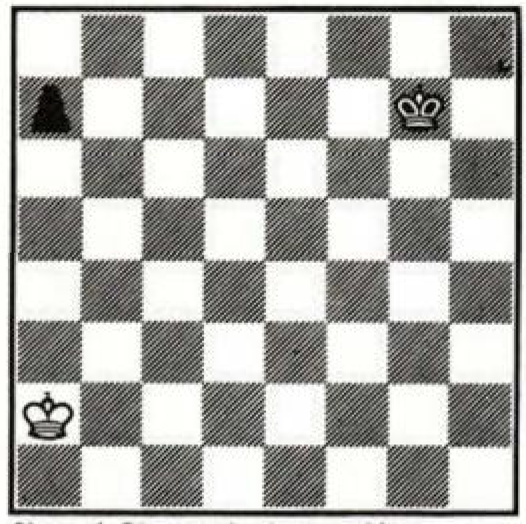


Figure 1. The classic chess problem.

This is combined with the first-ply score and compared with the highest total yet found for a program move, which is stored in location R(0). R(0) is continually updated as better moves are found for the machine — moves for which the opponent can find only weak responses.

An important feature of this search is the so-called "alpha-beta pruning". If any opponent response makes the machine's move under consideration worse than a previous stored machine's move, then there is no need for the machine to consider any further responses by the opponent to the machine move under consideration. The flag "AB" is set to 1, which stops any further searching of that move.

Alpha-beta pruning can save a good deal of unnecessary searching and thus a great deal of time. It is widely used in chess computers today. To be most effective, it is best to consider the most-likely-best machine move first, and also the most-likely-best opponent response.

The most-likely-best machine move has been derived by the sort which we considered earlier. The most-likely-best opponent move is hard to determine without going through them all — which, of course, defeats the whole point of alpha-beta searching.

I have instead adopted what I believe to be a novel heuristic: the best response found for the opponent for the previous machine move is evaluated first for the next machine move. This has proved to be quite effective at saving time in End-Game.

The nett effect of alpha-beta pruning on End-Game is quite spectacular in reducing response time. Anyone who doubts this should try deleting line 800.

The whole process I have described — move generation, sorting, counter-move generation, alpha-beta pruning — is known as iterative deepening. It will be appreciated that the best move so far found is always available, and machines using this technique generally display this best-move-yet — a feature which I have emulated in End-Game.

Many chess computers employ an adjustable timer which will interrupt the machine and display the best-move-yet as its move — examples include the Super System III — while others carry the process to its logical conclusion — examples are Sargon 2.5 and End-Game.

Because of the time restrictions imposed by interpreted Basic, End-Game evaluates the material and strategic position for both sides just once. Captures and certain other strong moves are evaluated for material gains at a further two levels.

The form of End-Game has been dictated by attempts to increase the speed. Constant calling of subroutines looks very pretty, but tends to slow execution time, while writing the same thing out several times is faster, but uses far more memory. I have stacked the most commonly-used subroutines at the head of the program to speed up their location when called.

The greatest retardation of any Basic program is caused by the dreaded If statement. When this occurs in a loop, the loss of time accelerates rapidly. The evaluation function is called after every potential move, yet If statements are essential in it if it is to serve any useful purpose. I have moved some of the evaluation features from the main evaluation subroutine to reduce the number of times they are called.

It is interesting to see how careful selection of moves can reduce total thinking time for the machine. Lines 130-170 are called every time a pawn move is considerd and should, one might think, slow the program compared with the speed of execution without these lines which test to see if advancing a pawn enables the opponent to snap it up immediately.

After all, the second level of search will find that the pawn can be captured by a strong opponent move, so why put it in? In fact, End-Game likes to advance pawns and so, by deterring an advance into the jaws of an opponent, a more sensible first move is put at the top of the list after the sort.

Thereafter, alpha-beta pruning does its work and the weak pawn advance is barely considered instead of being fully evaluated as the first move on the list. This saves a great deal of time. Thus the nett effect of the time-consuming lines 130-170 is actually to accelerate the program.

Variables defined in program lines 1110-1120:
CC = 0.012 CE = 0.1 CF = 0.2 CG = 0.3 CH = 10
CJ = 30 CK = 15 CL = 50 CM = 10 CQ = 2 CD = 1
CZ = 3

Material count: pawn = CD king = CZ	
Pawn moves:	
Do not approach enemy king	−CG
Do not approach enemy pawn	−CG
Stay off edge of board	−1/CH
Advance to rank Y	$+Y^2 \times CC$
Avoid having Y pawns on one file	$-(Y-1) \times Y \times CF$
Pawn advance: no opposition in first channel	+CG
to eighth rank	+0.5
score for first channel	Material count/CH
score for second channel	Material count/CM
En passant threat	−0.8
King moves:	
King opposition	+CE
King environment	+ 1/CK, + 3/CK
Avoid capture by pawn	−5
Do not stray from centres squares	+1/CJ
King ripple	Material count/(CQ × CL)

Table 1. Evaluation table for End-Game.

The evaluation features are listed in table 1. The variables which store the scores for different features, shown in table 1, are all found in lines 1110 and 1120, and so can be altered if you feel like experimenting.

Two features which I believe to be original are the pawn-advance routine — subroutine 510 — and the king ripple — lines 2260-2290. Both are stored outside the main evaluation subroutine.

The pawn-advance examines a three-square-wide channel ahead of the pawn after it has moved, all the way to the eighth rank. The move is scored according to whether the channel is obstructed — enemy piece in front, king scores high — or assisted — friendly piece in front, king scores high. The same channel is then examined again for its entire length, and again scored.

The second score shows whether the advancing pawn has numerical supremacy over the opposition: that is, one of two pawns will be encouraged to advance if the path is blocked by only one enemy pawn.

Obviously, this is a very crude evaluation feature, but it works relatively well for End-Game while minimising the number of If statements required.

The king ripple is a very low-scoring feature put in solely to prevent the king wandering aimlessly when most of the other material has been removed from the board. All the squares at a distance of two squares from the king, then three, then four and so on, are examined until another piece of either side has been found. The king then heads towards this piece.

King ripple is time-consuming and is evaluated only for the computer's pieces. Coupled with the routine which weakly discourages the king from wandering outside the central 16 squares, it should prevent the king from becoming "lost" for too long.

King environment searches each square within one move of the king, and scores favourably — + 3/15 — for each enemy pawn so located and less favourably — + 1/15 — for each friendly pawn. Obviously, the two kings cannot approach each other.

Other evaluation features include low scoring for pawns on either edge of the board, avoidance of doubled pawns — trebled or quadrupled pawns are punished exponentially — an exponentially-increasing score as a pawn advances to the eighth rank and encouragement for one king holding the opposition over the other.

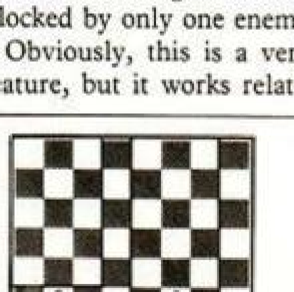
I have remembered End-Game's chess origins by not insisting that the machine advance a pawn to the eighth rank if a good move, such as a capture, exists elsewhere on the board.

En passant has been catered for by a somewhat elementary method. If the human makes a move which enables the machine to capture *en passant*, the capture is given priority and properly evaluated. However, the machine does not allow for *en passant* when otherwise evaluating moves: instead, the possibility of *en passant* is assigned a score of "undesirable" without evaluating in depth.

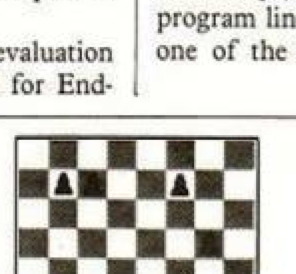
End-Game was written in standard Microsoft Basic with no Peeks or Pokes. The use of cursor commands, including screen clear and home greatly improves display.

Lines 1310 and 1770 operate a timer routine for my Sharp MZ-80K and can be adapted or ignored. Many computers do not like jumping from loops, which has influenced some of my program lines. Other Sharp users will require one of the Basic Extensions for the logical

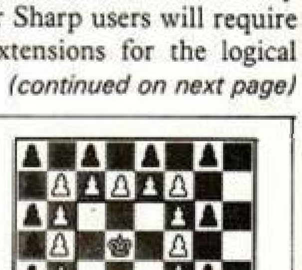
(continued on next page)



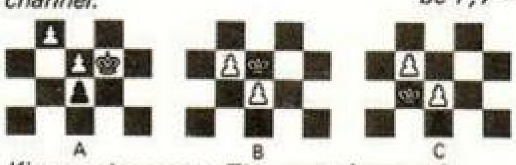
Pawn advance. A shows first channel, B second channel.



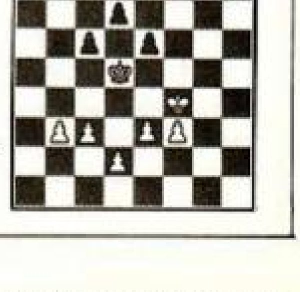
Pawn advance. The most likely move for black will be F,7 — F,6.



King ripple.



King environment. The most favoured position for the king is C; less favourable is B and least favourable is A.



A starting position for games pre-stored in End-Game.

(continued from previous page)

operators And and Or in some lines and the string inequalities in others.

Line 500 returns a value of −1 for each bracketed statement which is true, and 0 if false. This line runs some 20 percent faster than the corresponding If statements would.

Sadly, the program runs to 9.5K as it stands. This can be trimmed to 8K by removing the fast sort — this will slow it somewhat — by removing the screen-display lines, and by removing all but two of the data statements, together with the lines which select the data statements.

Program Function	Line
Evaluation	240-400
King environment	410-500
Pawn advance	510-629
Move storage	630-670
Third-level captures	680-810
Fourth-level captures	820-920
Data statements	930-1040
Variables defined for evaluation	1110-1120
Set-up position	1130-1280
First ply	1290-1540
Second ply	1550-1760
Move display	1770-1820
Input moves	1830-2080
King-move generator	2090-2310
Pawn one-move generator	2320-2350
Pawn two-move generator	2360-2450
Pawn capture	2460-2550
Fast sort	2560-2830
Alpha-numeric conversion	2840-2870
Screen display	2880-3010

End-Game has six different games prestored in 12 Data statements — the starting positions are shown in the diagrams. These can be selected, or the program will choose randomly between them. A display of the board is given — copy it on to your chess board. To set up your own position, it will be necessary to alter two Data statements.

The machine will prompt

YOUR MOVE

when read, followed by

FROM?

It will now accept ordinary alpha-numeric entries, such as

(FROM) D,2 (TO) D,4

Alternatively, typing P,1 will give a display of the board which is displayed only if you ask for it. Typing Q,1 will reveal what the machine thinks your move should be and typing Y,1 will cause the machine to act on its own suggestion for your move without need to enter it.

The only error check run by the machine on your input is that there is a piece of yours at the point from which you are trying to move. Thus you can move pieces round both easily and illegally should you want to.

```
99 REM**END-GAME by J.F.White.
100 PRINT[CLS]:GOTO1050
110 GOSUB230
120 GOSUB510:REM**CAN PAWN ADVANCE?
130 FORP3=-1TO1
140 IFA(X+P3,Y-AO)=-CZ*AOTHENO=O-AO*CG
150 IFP3=0THEN170
160 IFA(X+P3,Y-AO)=-AOTHENO=O-AO*CG
170 NEXT
180 N=N+1
190 GOSUB630
200 GOSUB680
210 GOSUB1480
220 RETURN
230 REM**EVALUATION
240 RX=A(X,Y):A(X,Y)=A(I,J):A(I,J)=0
250 O=0
260 FORIZ=1TO8
270 IFA(1,IZ)=AO OR A(8,IZ)=AOTHENO=O-AO/CH
280 FORJZ=1TO8
290 O=O+A(IZ,JZ)
300 IFA(IZ,JZ)=0THEN340
310 IFA(IZ,JZ)=CDTHENO=O+(9-JZ)*(9-JZ)*CC:MA=MA+1:GOTO340
320 IFA(IZ,JZ) =AO*CZGOSUB410:GOTO340
330 IFA(IZ,JZ)=-CDTHENO=O-JZ*JZ*CC:MB=MB+1
340 NEXT
350 O=O-((MA-CD)*CF*MA)
360 O=O+((MB-CD)*CF*MB)
370 MA=0:MB=0
380 NEXT
390 A(I,J)=A(X,Y):A(X,Y)=RX
400 RETURN
410 FORP3=-2TO2STEP4
420 IFIZ+P3<1ORIZ+P3>8ORJZ+P3<1ORJZ+P3>8THEN440
430 IFA(IZ+P3,JZ)=-CZ*AOORA(IZ,JZ+P3)=-CZ*AOTHENO=O+AO*CE:GOTO440
440 NEXT
450 FORI7=-1TO1:FORJ7=-1TO1
460 IFA(IZ+I7,JZ+J7)=0THEN480
470 O=O-((A(IZ+I7,JZ+J7)-2*AO)*CD/CK)
480 NEXT:NEXT:O=O+3*AO/CK
490 IFA(IZ+CD,JZ-AO)=-AOORA(IZ-CD,JZ-AO)=-AOTHENO=O-5*AO
500 O=O+((IZ<3)+(IZ>6)+(JZ<3)+(JZ>6))*AO/CJ:RETURN
510 FORP2=-1TO1:IFP2+X>8THEN580
520 FORO2=Y-AOTO(4-4*AO)STEP-AO
530 OP=OP+A(P2+X,O2)
540 NEXT
550 FORO2=1TO8
560 OO=OO+A(P2+X,O2)
570 NEXT
580 NEXT
590 IFOP=0THENO=O+CG*AO
600 IFY=1THENO=O+.5
610 O=O+OP*CD/CH+OO/CM:OP=0:OO=0
620 RETURN
630 REM**MOVE STORAGE
640 AA(N)=X:BB(N)=Y
650 AB(N)=I:BC(N)=J
660 D(N)=O
670 RETURN
680 REM**EVALUATE CAPTURES
690 IFQA(K)-O(N)>.8OANDFLAG=1THEN710
700 GOTO790
710 REM**3RD LEVEL
720 FORI5=-1TO1:FORJ5=-1TO1
730 IFI5=0ANDJ5=0THEN750
740 IFA(AA(N)+I5,BB(N)+J5)=CZ*-AOTHENO(N)=O(N)-A(AB(N),BC(N)):GOSUB830:NG=1
750 NEXT:NEXT
760 IFNG=1THENNG=0:GOTO790
770 IFA(AA(N)+AO,BB(N)-AO)=-AOTHENO(N)=O(N)-A(AB(N),BC(N)):GOSUB830:GOTO790
780 IFA(AA(N)-AO,BB(N)-AO)=-AOTHENO(N)=O(N)-A(AB(N),BC(N)):GOSUB830
790 IFO(N)<SOTHENSO=O(N):S1=AA(N):S2=BB(N):S3=AB(N):S4=BC(N)
800 IFQA(K)+SO<=R(O)THENAB=1
810 RETURN
820 REM**FOURTH LEVEL
830 FORI4=-1TO1:FORJ4=-1TO1
840 X2=AA(N)+I4:Y2=BB(N)+J4
850 IFX2=AB(N)ORY2=BC(N)THEN870
860 IFA(X2,Y2)=CZ*AOTHENO(N)=O(N)+AO:NG=1
870 NEXT:NEXT:IFNG=1THENNG=0:RETURN
880 IFAA(N)-1=AB(N)THEN910
890 IFA(AA(N)-1,BB(N)-1)=AOTHENO(N)=O(N)+AO:RETURN
900 IFAA(N)+1=AB(N)THEN920
910 IFA(AA(N)+1,BB(N)-1)=AOTHENO(N)=O(N)+AO
920 RETURN
930 DATA4,5,3,2,6,1,3,6,1,4,7,1,5,6,1,6,6,1
940 DATA 6,4,-3,2,3,-1,3,3,-1,4,2,-1,5,3,-1,6,3,-1
950 DATA 4,6,3,1,7,1,2,7,1,3,7,1,7,7,1,8,7,1
960 DATA7,3,-3,1,2,-1,2,2,-1,3,2,-1,6,2,-1,8,3,-1
970 DATA4,6,3,1,5,1,2,7,1,8,7,1,0,0,0,0,0,0
980 DATA4,4,-3,2,3,-1,7,3,-1,8,3,-1,0,0,0,0,0,0
990 DATA4,8,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
1000 DATA4,4,-3,4,2,-1,0,0,0,0,0,0,0,0,0,0,0,0
1010 DATA7,6,3,8,6,1,0,0,0,0,0,0,0,0,0,0,0,0
1020 DATA6,3,-3,7,3,-1,8,3,-1,0,0,0,0,0,0,0,0,0
1030 DATA4,6,3,2,6,1,3,5,1,5,6,1,7,7,1,8,6,1
1040 DATA5,2,-3,1,2,-1,3,4,-1,4,2,-1,7,3,-1,8,3,-1
1050 DIMAA(25),BB(25),D(25),A(20),B(20)
1060 DIMC(20),D(20)
1070 DIMA(9,9)
1080 DIMAB(25),BC(25),QA(25)
1090 DIMR(4),S(4),R$(4),S$(4)
1100 DIMB(INT(LOG(25)/LOG(2)+1),2)
1110 CC=.012:CD=1:CE=.10:CF=.2:CG=.3:CH=10: REM **VARIABLES IN EVALUATION
1120 CI=10:CJ=30:CK=15:CL=050:CM=10:CZ=3
1130 PRINT:PRINT:PRINT
1140 PRINT"SELECT POSITION(1-6) OR RND"
1150 INPUTA$:IFA$="RND"THENRN=INT(RND(1)*6)+1:GOTO1170
1160 RN=VAL(A$):IFRN<1ORRN>6THEN1140
1170 PRINT[CLS][8CD][5CR]"END - GAME by J.F.White."
1180 FORIA=1TORN
1190 FORI=1TO8:FORJ=1TO8:A(I,J)=0:NEXT:NEXT
1200 FORI=1TO16:IFI<11ANDI>6THEN1220
1210 READA(I),B(I),A(A(I),B(I))
1220 NEXT
1230 NEXT
1240 RESTORE
1250 GOSUB2880
1260 PRINT:PRINT"DO YOU WANT TO GO FIRST(Y\N)?"
1270 INPUTA$:IFA$="Y"THENPRINT:PRINT:GOTO1830
1280 IFA$<>"N"THEN1260
1290 PRINT[CLS][2CD]"   FROM    ,    TO    , "
1300 TO=-100
1310 TI$="000000"
1320 R(O)=-100
1330 AO=1:SO=99:AB=0
1340 FLAG=0
1350 GOTO1390
1360 I=S(3):J=S(4):SO=99:AB=0
1365 IFK=1THEN1390
1370 IFA(I,J)=AOGOSUB2330
1380 IFA(I,J)=CZ*AOGOSUB2090
1390 FORI=1TO8:FORJ=1TO8
1400 IFAB=1THEN1430
1410 IFA(I,J)=CZ*AOGOSUB2090
1420 IFA(I,J)=AOGOSUB2330
1430 NEXT:NEXT
1440 IFN<>0THEN1540
1450 PRINT:PRINT:PRINTTAB(7);"STALEMATE !":PRINT:END
1480 IFFLAG=1ORN=0THEN1530
1490 IFD(N)>TOTHENTO=D(N):T1=AA(N):T2=BB(N):T3=AB(N):T4=BC(N):GOTO1510
1500 GOTO1530
1510 T1$=CHR$(ASC(STR$(T1))+16):T3$=CHR$(ASC(STR$(T3))+16)
1520 PRINT[HOME][2CD]";TAB(9);T3$;TAB(12);T4;TAB(20);T1$;TAB(23);T2
1530 RETURN
1540 IFFLAG=1THENRETURN
1550 REM**SORT
1560 M=N:FLAG=1:AO=-1
1570 GOSUB2560
1580 REM**2ND LEVEL
1590 R(O)=-99
1600 FORK=1TOM
1610 A(M-K+1)=AA(K):B(M-K+1)=BB(K):C(M-K+1)=AB(K):D(M-K+1)=BC(K)
1620 QA(M-K+1)=D(K):D(K)=0
1630 AA(K)=0:AB(K)=0:BC(K)=0:BB(K)=0
1640 NEXT
1650 FORK=1TOM
1660 N=0
1670 RY=A(A(K),B(K)):A(A(K),B(K))=A(C(K),D(K)):A(C(K),D(K))=0
1680 GOSUB1360
1690 OZ=SO+QA(K)
1700 IFOZ>R(O)THENR(O)=OZ:R(1)=A(K):R(2)=B(K):R(3)=C(K):R(4)=D(K):GOTO1720
1710 GOTO1750
1720 S(1)=S1:S(2)=S2:S(3)=S3:S(4)=S4
1730 GOSUB2840
1740 PRINT[HOME][2CD]TAB(9);R$(3);TAB(12);R(4);TAB(20);R$(1);TAB(23);R(2)
1750 A(C(K),D(K))=A(A(K),B(K)):A(A(K),B(K))=RY
1760 NEXT
1770 PRINT"MY TIME FOR THIS MOVE = ";MID$(TI$,3,2);" MINS ";RIGHT$(TI$,2);
1772 PRINT" SECS."
1780 GOSUB2840
1790 PRINT[3CD][2CR]"FROM ";R$(3);" ,";R(4);" TO ";R$(1);" ,";R(2)
1800 IFR(2)=1 ANDA(R(3),R(4))=1THENPRINT:PRINT:PRINT[8CR]"I WON!":END
1810 IFABS(R(1)-R(3))=1ANDR(2)=3ANDA2=R(1)ANDB2=R(2)+1THENA(A2,B2)=0
1820 A(R(1),R(2))=A(R(3),R(4)):A(R(3),R(4))=0
1830 PRINT:PRINT"YOUR MOVE? " :FLAG=0:N=0:TO=-100
1840 PRINT: INPUT"FROM  "; A1$,B1$:B1=VAL(B1$)
1842 IFA1$="Y"THENA1$=S$(3):B1=S(4):A2$=S$(1):B2=S(2):GOTO1900
1850 IFA1$="P"THENGOSUB2880:GOTO1830
1860 IFA1$<>"Q"THEN1890
1870 PRINT"YOUR BEST REPLY IS ";S$(3);" ,";S(4);" TO ";S$(1);" ,";S(2)
1880 GOTO1830
1890 INPUT"TO  ";A2$,B2$:B2=VAL(B2$)
1900 A1=VAL(CHR$(ASC(A1$)-16))
1910 A2=VAL(CHR$(ASC(A2$)-16))
1920 IFA(A1,B1)<>-1THENPRINT"ERROR - TRY AGAIN":GOTO1830
1930 IFB2=8ANDA(A1,B1)=-1THENPRINT:PRINT:PRINT[3CR]"YOU WON!":PRINT:END
```

```
1940 PRINT[CLS][2CD]"   FROM   .   TO   . "
1950 A(A2,B2)=A(A1,B1):A(A1,B1)=0
1960 IFA(A2,B2)=-1ANDB1=5THEN1980
1970 GOTO1990
1980 IFABS(A1-A2)=1ANDR(2)-R(4)=-2ANDR(1)-A2=0THENA(A2,B2-1)=0
1990 IFB1=2ANDB2=4THENA0=1:GOTO2010
2000 GOTO2080
2010 IFA(A2+1,B2)=1THENI=A2+1:J=B2:N=1:GOSUB2040
2020 IFA(A2-1,B2)=1THENI=A2-1:J=B2:N=1:GOSUB2040
2030 GOTO2080
2040 A(A2,B2-1)=A(A2,B2):A(A2,B2)=0:GOSUB2470:A(A2,B2)=A(A2,B2-1)
2050 A(A2,B2-1)=0
2060 TO=O(N)
2070 RETURN
2080 GOTO1310
2090 REM**KING MOVE
2100 FORII=-1TO1:FORIJ=-1TO1
2110 IFAB=1THEN2300
2120 IFII=0ANDIJ=0THEN2300
2130 X=I+II:Y=J+IJ
2140 IFX <1ORX >8ORY <1ORY >8THEN2300
2150 IFA(X,Y)=AOTHEN2300
2160 FORF4=-1TO1:FORD4=-1TO1
2170 IFA(X+F4,Y+D4)=-CZ*AOTHENNG=1
2180 IFA(X-1,Y-AO)=-AOTHENNG=1
2190 IFA(X+1,Y-AO)=-AOTHENNG=1
2200 NEXT:NEXT:IFNG=1THENNG=0:GOTO2300
2210 GOSUB230
2220 N=N+1
2230 GOSUB630
2240 GOSUB680
2245 IFFLAG=1THEN2300
2250 OG=0:CO=2
2252 FORP3=-COTOCO
2254 IFX+P3<1ORX+P3>8THEN2264
2256 IFY+CO>8THEN2260
2258 OG=OG+ABS(A(X+P3,Y+CO))
2260 IFY-CO<1THEN2264
2262 OG=OG+ABS(A(X+P3,Y-CO))
2264 IFY+P3<1ORY+P3>8THEN2274
2266 IFX-CO<1THEN2270
2268 OG=OG+ABS(A(X-CO,Y+P3))
2270 IFX+CO>8THEN2274
2272 OG=OG+ABS(A(X+CO,Y+P3))
2274 NEXT
2276 IFOG>0THENO(N)=O(N)+OG/(CO*CL):GOTO2290
2278 CO=CO+1:IFCO>8THEN2290
2280 GOTO2252
2290 GOSUB1480
2300 NEXT:NEXT
2310 RETURN
2320 REM**PAWN 1-MOVE
2330 X=I:Y=J-AO
2340 IFA(X,Y)<>0THEN2460
2350 GOSUB110
2360 REM**PAWN 2-MOVE
2370 IF(AO=1)AND(J=7)THEN2400
2380 IF(AO=-1)AND(J=2)THEN2400
2390 GOTO2460
2400 X=I:Y=J-2*AO
2410 IFA(X,Y)<>0THEN2460
2420 GOSUB110
2425 REM**EN PASSANT THREAT
2430 IFA(X+1,5+(AO*-CD))=-AOTHENO(N)=O(N)-AO*.8
2440 IFA(X-1,5+(AO*-CD))=-AOTHENO(N)=O(N)-AO*.8
2450 GOSUB1480
2460 REM** PAWN CAPTURE
2470 X=I-1:Y=J-AO
2480 IF(AO=1)AND(A(X,Y)>=0)THEN2510
2490 IF(AO=-1)AND(A(X,Y)<=0)THEN2510
2500 GOSUB110
2510 X=I+1:Y=J-AO
2520 IF(AO=1)AND(A(X,Y)>=0)THEN2550
2530 IF(AO=-1)AND(A(X,Y)<=0)THEN2550
2540 GOSUB110
2550 RETURN
2560 REM**QUICK SORT
2570 SS=1
2580 B(1,1)=1:B(1,2)=M
2590 LL=B(SS,1):RR=B(SS,2):SS=SS-1
2600 II=LL:JJ=RR:XX=O(INT(RND(1)*(RR-LL)+.5)+LL)
2610 IFO(II)>=XXTHEN2630
2620 II=II+1:GOTO2610
2630 IFXX>=O(JJ)THEN2650
2640 JJ=JJ-1:GOTO2630
2650 IFII>JJTHEN2720
2660 WW=O(II):O(II)=O(JJ):O(JJ)=WW
2670 WW=AA(II):AA(II)=AA(JJ):AA(JJ)=WW
2680 WW=BB(II):BB(II)=BB(JJ):BB(JJ)=WW
2690 WW=AB(II):AB(II)=AB(JJ):AB(JJ)=WW
2700 WW=BC(II):BC(II)=BC(JJ):BC(JJ)=WW
2710 II=II+1:JJ=JJ-1
2720 IFII<=JJTHEN2610
2730 IFJJ-LL>=RR-IITHEN2770
2740 IFII>=RRTHEN2760
2750 SS=SS+1:B(SS,1)=II:B(SS,2)=RR
2760 RR=JJ:GOTO2800
2770 IFLL>=JJTHEN2790
2780 SS=SS+1:B(SS,1)=LL:B(SS,2)=JJ
2790 LL=II
2800 IFLL<RRTHEN2600
2810 IFSS>0THEN2590
2820 PRINT"SORT COMPLETE"
2830 RETURN
2840 FORKI=1TO35STEP2
2850 R$(KI)=CHR$(ASC(STR$(R(KI)))+16) :S$(KI)=CHR$(ASC(STR$(S(KI)))+16)
2860 NEXT
2870 RETURN
2880 REM**SCREEN DISPLAY OF POSITION
2890 PRINT[CLS]
2900 FORJ=8TO1STEP-1:FORI=1TO8
2910 PRINTTAB(5*I-5);
2920 IFA(I,J)=0THENPRINT".";
2930 IFA(I,J)=CDTHENPRINT"P";
2940 IFA(I,J)=-CDTHENPRINT"P*";
2950 IFA(I,J)=CZTHENPRINT"K";
2960 IFA(I,J)=-CZTHENPRINT"K*";
2970 NEXT:PRINT J :PRINT:NEXT
2980 PRINT:PRINT"A    B    C    D    E    F    G    H"
2990 PRINT:PRINT:PRINT"PRESS ANY KEY TO CONTINUE"
3000 GETA$:IFA$=""THEN3000
3010 RETURN
```

Writing chess

Drawing on examples from his program written for the Sinclair machines, Philip Joy shows you how to go about creating a chess game of your own.

THE ZX-80 IS not the best machine on which to write long and complex programs because of its Basic. There are hardware problems as well as software ones — a poor keyboard for entering code in quantity, and slow speed. The Basic does not let you use two-dimensional arrays — which at first might seem a problem — because of an eight-by-eight board.

However, it proves not to be, and in fact it helped me to such an extent that I have kept a 64 array for my version for the ZX-81, even though it has multi-dimensional arrays. By having integer arithmetic only, the troubles with many INTs were overcome. The main hardship is the fact that it did not have a really easy way to enter machine code. Read or Data would have helped or even a monitor would have made the entry of machine code easier.

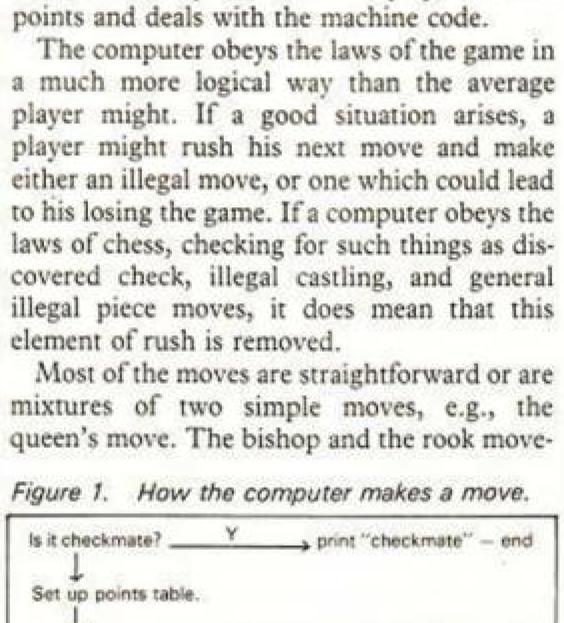
As the program neared completion, it was structured around five main units: initialisation, movement, points, player, and back-up. Each one of these units has a specific use and place in the program. The movement was the most difficult to write and proved to have a great deal of bugs. The points section is the thinking part of the program and calculates the best moves: it was the easiest part to write.

The player is the unit which keeps the user informed and sorts his moves for the computer to respond to. Back-up plays the greatest part in keeping the program working smoothly. It finds, for example, the level of play, sorts the points and deals with the machine code.

The computer obeys the laws of the game in a much more logical way than the average player might. If a good situation arises, a player might rush his next move and make either an illegal move, or one which could lead to his losing the game. If a computer obeys the laws of chess, checking for such things as discovered check, illegal castling, and general illegal piece moves, it does mean that this element of rush is removed.

Most of the moves are straightforward or are mixtures of two simple moves, e.g., the queen's move. The bishop and the rook movements could be calculated by a person with some experience of computer programming. With some thought, the L-shaped knight movement and the queen's could be solved. However, the two pieces, the king and the pawn which move only one square at a time, are the pieces with many conditions attached to them. These two pieces have many characteristics which, although are not directly connected with them, make the movement hard to perfect.

Figure 1. How the computer makes a move.



The program is based around a 64-character array which holds all the black and white pieces and the blank spaces. The program uses two vectors to search for its move. One is the horizontal vector, or the number of squares down, while the vertical or the number of squares along is used as the other vector. To look into the array, these two vectors are brought together by the equation

$$(x-1) * 8 + y$$

to give the square about which we want to find some information. Our points table is scanned for a move, and this move is then tried. If the move fails, it is given a value of 0 in the points table to prevent its being tried again. When a move is found, the array is updated, and the board is printed for the player's turn. In any move, the important thing to remember is that the moving of a piece can cause check, or checkmate. Many human players could overlook this part of the game.

(continued on next page)

(continued from previous page)

After each move, the computer will discover whether its king is in check. If it is, the position is restored and another move tried. If the same directions for a piece were tried in the same sequence every time, the bishop, for example, would always move upwards and left. The computer must, therefore, try different directions in a different order each time.

Difficult decisions

The initialisation stage is the section where the most difficult decisions are made. For example, you have to choose the form of storage, the arrays and their sizes and the different variables. The representation of the pieces, and the style of play will all be affected by the way you decide. It is also here where you can include features which are not needed or could be combined to reduce space, and time.

The movement of the pieces is the only part of the program which has been heavily flow-charted. It is logical and is a part humans do not consider in great detail — hence the danger of bugs. It took about a month to remove the bugs in this section, mostly by trial and error.

We search for a possible move by working across the board and using our memory to decide whether a piece can move or not. The computer discovers a move by trial and error. We must also be careful to take into account that the program could be faced with a situation for which it cannot find a move. In this position, the computer will have to decide whether it is checkmate or stalemate.

The pawn was the easiest to tackle with its one move forward or two if it was on its first move. All you have to do is to subtract from the horizontal vector and check whether the piece can move to this square. If it is on row 7 or row 2 depending on colour, we can take two from the horizontal vector. However, before we move we can check whether a diagonal — found by adding or subtracting one from both the vectors, depending on which way you are going — is occupied by an enemy piece. If it is, we would take it if it does not put the computer's king in check. The only problem with the pawn arises when it is about to check — it has two possible moves.

The bishop and the rook are roughly the same. They created no problems and took very little time to develop. For a rook, we add one to the file or row depending on which direction it is moving. The bishop is slightly more difficult as it has to add or subtract one from the file and row — again depending on which way it was moving. When we have found our new position, and before we make it on the board, we check for a number of things.

Complicated pieces

First, we see whether it has reached the other side, or whether it has reached a piece of its own colour. In either case, we know that this is as far as it can go. If it reaches an opponent's piece, it can replace it with its own colour and subtract one from the number of pieces the opponent has on the board. We must also check for a check or checkmate and if there is one, we must either move it to another square or not move that piece at all.

The knight, although it may seem a complicated piece, is very similar to the rook and bishop. You add two to one direction and take one from the other, or any other combination, and you must remember that this piece only moves once, unlike the rook. There are in fact eight moves a knight can make and you must check for the piece going off the board.

The king can move in any direction but only one square at a time, so you can see how easy it is to cater for it in the program. We must remember, however, that it is the piece which must not be attacked. We have to find out whether it is in check both at the beginning, and at the end.

The way this is done is to search along every diagonal, file and row until we meet an edge or a piece of the same colour. If this happens, move on to the next path because the king is not being attacked in that direction. A simple For-Next loop will deal with that.

However, it we find an opponent's piece we must discover which piece it is since all pieces, apart from the queen, are limited in their directions of attack. If the piece is not an attacker, we can consider knight moves away from the king to see if they contain a knight of the opposite colour. If we do not find check then we can move on.

Element number	Value	Comments
1	0	piece either not present or has been tried
2	540	a pawn on its starting square
3	600	a queen being attacked
4	539	a pawn being attacked
5	450	a bishop in the middle, not being attacked
6	460	a castle on its starting square
7	440	a pawn
8	440	a pawn
9	443	a pawn — a random number is added to make sure each game is different
10	510	a knight in a poor position
11	580	a knight being attacked
12	340	the king
13	0	a piece already tried
14	508	a bishop in a poor position
15	570	a pawn on the square before it is promoted
16	0	a piece already tried

Table 1. Points table with list of pieces.

If we find check, we have the other problem of discovering whether it is checkmate. The three ways of eluding check are: moving the king, taking the attacking piece, or blocking. The first is the easiest — move the king to all the possible positions and verify to see if the king is in check. If he is not in check, the situation has been solved.

The next option — taking the attacker — is a far more difficult problem to solve. We must use the same routine to see if any piece is attacking. If it is, we can move it to take the attacker and, as long as we do not cause another check or double check, we have again solved the situation.

Blocking a check with a piece is the most difficult aspect. We can first see if the attacker is a knight, because if it is we cannot block. If it is not, we must, using a For-Next loop, go from every square between the king and the attacker and see if any one of those squares is attacked by one of its own pieces. If it is, we can move the piece to block the attacker.

This again has two conditions, the first is that it is not a double attack, and the second that it does not create a discovered check. If you still cannot find a way out, the computer is in checkmate, and you should have it say so.

The part which decides how well a computer plays is its points table. Each program has its own points' table and a different way of filling it.

The points table is made up of 16 elements, and each element is initially given 500 points. The computer then goes through the board until it finds one of its pieces; it then stops and evaluates the position. It does this by looking at which piece it is and subtracts points if it does not want to move it.

Skill of the game

For example, it is much safer to move a bishop than a queen. The program considers other aspects such as whether it is being attacked. If it is, it will have to move the piece concerned so some points will need adding. Then it can look at its position; if it is in the centre, points will be deducted so that it is less likely to move. Other considerations can be evaluated, depending on how good a game you wish to play, and how much space you have. When this points table is built up, the element with the most points will be tested to see if it can be moved. If it cannot then a zero will be placed in it so that it will not be tested again.

The element with the most points moves first. The skill and standard of the game depends so much on this part of the game that to make it play better, a great deal of work on this section is needed. The results we obtain from a points table need not be accurate enough for playing against some players, but it can be a match for average players if it is dealt with correctly.

The player section is the part which will interact with the player so that he can enter his moves and can see the board. My program displays a full board with the pieces represented as letters:

King-K, Queen-Q, Rook-C, Bishop-B, Knight-N, and Pawn-P.

This is satisfactory for a ZX-80 but if your computer has graphics, or else a user-definable character set, then use the letters. The normal way for entering chess programs into a computer such as a Chess Challenger is by algebraic methods. That is used here — it is also the chess standard.

The final section of the game is the back-up which is just a collection of routines which will deal with such things as:

- Set up points table
- Zeroing of moves already tried
- Loading and using machine code if any is present.
- Reprinting board.
- Putting pieces on the board.
- Different board set-ups — black or white.
- Other tasks which can be used by all of the routines already mentioned. ■

A & F SOFTWARE

MORE NEW REAL TIME GAMES FOR ATOM USERS

G0100 **POLECAT:** Avoid being eaten by the polecat searching the maze for you. 5K text 6K graphics.............................£4.95

G0101 **EARLY WARNING:** Destroy the attacking ICBM's using a realistic radar display and intercept missiles. 48 levels. Sound. Score and screen counters. 4K text 6K graphics............£4.95

G0102 **MINEFIELD:** Watch out for chain-reactions as you clear the mines with your tank. Each mission becomes harder! Sound. Score and high score. 5K text ½ K graphics.................£4.95

G0103 **TANGLED:** Challenge your friend or the Atom to this game of skill. The more you score the harder you find the game. Four skill levels. Sound. Individual and highest scores displayed. 5K text 6K graphics..£3.95

E0200 **TIME TUTOR:** Teach your children the time the modern way, the program randomly selects a time and displays it on a standard clock face, responding to the students input telling the correct time if the student is wrong. Makes learning fun. Sound. Score. 5K text ½ K graphics......................................£3.95

*** SPECIAL OFFER *** (supplied on one tape)

Select any two programs from "Polecat", "Early Warning" or "Minefield" for only £8.50 or all three for only £10.00

All four games programs for only £12.00

"Dealer enquiries welcome"

Order Form: Send cheque/PO payable to: **"ANDERSON" to A&F SOFTWARE, 10 WILPSHIRE AVE., LONGSIGHT, MANCHESTER M12 5TL (061 248 7195)**

Return name and address

Code No.	Program Title	Qty	£	p

(S.A.E. for full list of available programs) Special Offer Please order on separate sheet Total

---

Had enough of games?

The Sinclair ZX81

PROGRAMMING FOR REAL APPLICATIONS

A NEW BOOK BY RANDLE HURLEY
PUBLISHED BY THE MACMILLAN PRESS

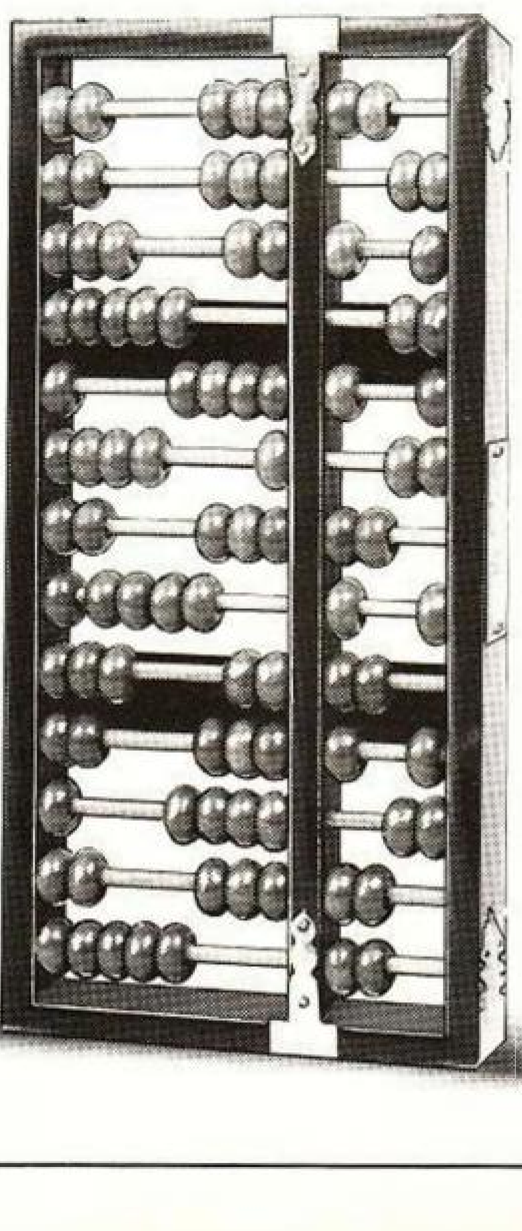
Yes, the 16K ZX81 *can* be used for serious applications. Here is a book that tells you how to make the best use of the 16K ZX81, how to write big, serious programs in BASIC. These programs are all fully described, listed, and documented:

WORD JUGGLER – a word processor with cursor control, delete, edit, reading age calculator and space for 11,000 characters. **SAFE INPUT ROUTINE** to avoid input crashes. **CASH HANDLING SYSTEM. PERSONAL FINANCES** to look after your accounts – cheques, standing orders, credits, etc. **BANKING,** designed to run a school bank. **RECORD** stores 18,000 pieces of information – more than the number of bytes in RAM! **RANK ORDER** and **EXAM STATISTICS** for analysing school results (DES requirements). And much, much, more!

£7.35 for 180 pages, including postage and packing

Order from Liz Digby Firth, Globe Book Services, Houndmills, Basingstoke, Hants.

CASSETTE of all the above programs – saves hours of keying! Requires 16K ZX81 and is useless without the book . . . 30 mins. of data for just £11.95 including V.A.T. and postage.



START *low cost* COMPUTING IN CAMBRIDGE

For less than £100 you can launch yourself into the world of computing with your own system. At Cambridge Computer Store our "Budget Micro" department offers an exceptionally wide range of inexpensive machines, all generally on demonstration, available from stock and fully supported by our enthusiastic staff.

- **Acorn Atom**
- **Commodore VIC 20**
- **Tandy Colour Computer**
- **TRS-80 Model I**
- **UK 101 kit system**
- **Sharp Pocket Computer**
- **Low-cost peripherals**

Make the right start! Visit us at:

Cambridge Computer Store

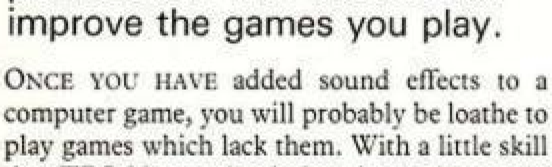
1 Emmanuel Street, Cambridge CB1 1NE
Telephone (0223) 65334/5

SOUND VIEWS ON
IN TUNE WITH TANDY

Alun Evans' nine assorted — and easily converted — Basic and assembler routines are designed to reveal the musical side of the TRS-80's nature. The sound effects they produce can also be used to improve the games you play.

ONCE YOU HAVE added sound effects to a computer game, you will probably be loathe to play games which lack them. With a little skill the TRS-80 can be induced to gloop and gurgle like the very best arcade Space Invaders and can be made to play tunes and to respond to the keyboard like a simple electronic organ.

The bad news, for Basic fans, is that the means to make sounds is through machine-code programs. Before you abandon hope, however, take heart: I have supplied the data



Figures 1a and 1b.

that sets up the machine code so that the routines can be Poked into the top memory and accessed from Basic.

Those who understand assembly language will find some suggestions for further refinements of the routines so that they can be more widely used.

Sound, a simple pure note, is a sine wave — figure 1. The pitch of the note is a function of the rate at which the cycles are produced. The more cycles produced in a given time, the higher the pitch of the note. Figure 1a shows a note which has a higher pitch — more cycles — than the note in figure 1b. The duration of the note is the length of time we allow a particular rate of cycles to be produced.

It is these two variables, Pitch and Duration, which we shall have to control when we make sounds with the computer.

Sound is generated in the cassette interface which watches bits 0 and 1 at port 255. The since wave of a simple note is approximated to, in the TRS-80, as shown in figure 2. We set port 255 to 1, hold it at 1 and then reset it to 2. If we keep doing this we end up with a square wave-form.

The length of time we hold the port in either state is shown as "f". Clearly, the longer we make "f", the fewer cycles we shall be able to complete in a given time. Hence the longer "f" is, the lower the pitch of the note. Similarly, we raise the pitch by shortening "f".

In the Radio Shack-published *TRS-80 Assembly-Language Programming* by Barden, there is a simple assembler program which enables us to produce square waves and which gives us control over both pitch and duration of the note.

However, before we start on the production of some sound-making code, we must set the machine so that we can hear what we are doing. Connect the larger of the two grey jack-plugs which normally plug into the computer's CTR into the input socket of an amplifier. I use an old Sanyo cassette radio recorder, plugging into the microphone socket.

I then put a dummy tape into the machine, set it to record, and hear the computer via the recorder's monitoring facilities. Remember to replace the jack-plug into the computer-controlled CTR before attempting any CSaving — or you will finish with a blank tape.

Now let us examine Barden's routine — listing 1. Notice that zero values are given for both pitch and duration. This is because they will be Poked into the routine from Basic.

Listing 2 shows the Basic which will let you experiment with tones produced by the machine code. All you obtain at this stage is a single tone repeated 10 times. However, if you keep a record of the numbers you have tried, you begin to develop a feel for the kind of noise to expect from various values.

Experimenting with listing 2 should show you that the relationship between the duration of a note and its pitch is a complex one. If you hold the value for duration constant and vary the pitch, you will see that the higher notes play for less time than the lower ones.

Even such rudimentary sound facilities have their uses. Listing 3 demonstrates just how easy it would be to make your favourite games more lively. Note that there is no need to

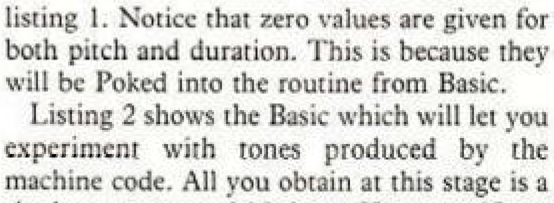


Figure 2.

repeat the Mem Size fix or the lines that Poke in the machine code unless the computer has been turned off between programs.

The next step is to experiment with tunes stored in data lines. This is what the listing 4 does. Once again, repeat only lines 10-60 from listing 1 if you have corrupted any of the contents of the protected memory space.

Saving the musical data as an array, as in lines 90-110, has two advantages. First, it avoids the need to reread the data if we want to play the tune more than once. Do not forget that Restore would reactivate all data in a program, not just the piece we want. Secondly, it makes it possible to rearrange the order in which the notes are played. Delete lines 130-180 from the listing 4 and add listing 5. The computer is now composing, and playing, its own tunes.

To make the computer into a simple organ, we need a program that watches the keyboard for key-pushes, and which then supplies pitch values to the sound-making routines. Inkey$ would do this but that would supply only a one-off burst of sound — no bad thing if you wanted an audio response to be given to the user but not what we want for the organ. Better for our purposes would be to Peek the keyboard.

The TRS-80s operating system locates the number keys 0-7 at address 12448. By Peeking this address the program can read which of the keys is being pressed. If

```
PEEK (12448) = 1
```

then the key is 0. If

```
PEEK (12448) = 2
```

Melody

The sound of MZ-80K is something owners of the machine can enjoy without having to buy a costly sound-box. Bob Edwards reports.

THE SHARP MZ-80K computer is remarkably versatile in many respects. For instance, it has a set of graphics superior to most other computers in its range. Unlike some of its near competitors, most of the graphics are accessible directly from the keyboard — more than 104 shapes at the press of a key.

Another way in which the MZ-80K is ahead of its rivals is in its built-in music capability; there is no sound box to buy as an optional extra. A straightforward statement such as:

```
10 Music"# B1A1R4A1# A1-C-AR1-A3-F3R5"
```

will give a passable rendering of the first line of *Colonel Bogey.* Making this line into a string by replacing the word Music with M$ = makes life even easier;

```
20 MUSIC M$
```

will make the machine play the entire phrase. Using the same principle to produce single notes of varying length can provide the beeps and boops that make computer game-playing that much more fun.

This facility, especially when combined with the excellent graphics set, means you can create computer games which surpass those of other machines in the price range. The budding Beethovens among us can compose tunes to their heart's content and then hear them played back at once.

MAKING MUSIC

then the key is 1 and so on up to a maximum

PEEK (12448) = 128

for key 7.

Listing 6 is the Basic which will give you the Stylophone you always wanted.

Assembly-language buffs will see that it would be far more efficient and relatively simple to have the machine language watch the keyboard, rather than Basic. The increase in speed should make it possible to watch all the keys and should remove the *vibrato* effect from the notes.

Sound effects, laser cannon, flying saucers and the like, all depend on the sound-making routine being supplied with a sequence of frequency values. Each of these it plays for a very short duration and there should be no noticeable gap between the notes used.

The machine-language routine we have been using so far has to return to Basic to find what, if any, values we want it to use next. This process is too slow to achieve the kinds of effects we want.

The answer is to rewrite the machine code so that the routine feeds itself from a block of data that we have set up for it in advance. The assembler listing is reproduced in listing 7.

Basic programmers will find this routine translated into Pokes in listing 8. Before using the Basic though, look at figure 3.

This shows the TRS-80 memory area where the new routine lies. Notice that the data used by the routine — the frequencies we will want it to play — lie immediately above the routine itself. If you want to use another set of notes, or want to try out sequences of sounds you have invented, it is into this area, from 32489 onwards, that you must Poke your data.

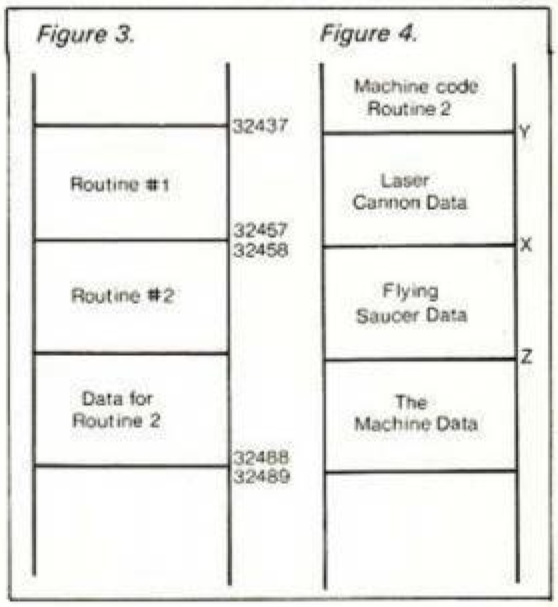
Note, too, that all such sound sequences must end with a zero. It is the 0 that turns off the sound and returns you to Basic. Omit it and you will probably have the humiliation of groping for the reset switch to obtain silence.

Listing 9 offers some replacement sound sequences. In each case replace line 50 in listing 8.

The important thing to realise when you begin to experiment with patterns of your own is that the duration value is fixed for the whole sequence. That means high-pitched notes will play for relatively much less time than the lower ones.

As a counter measure to this, I have found you need to pack the sequences with many high-frequency values, decrementing slowly, while the deeper notes can be fewer in number and can make bigger jumps in value. You could create a table of duration values, and

(continued on next page)



of Sharp distinction

I use the line from *Colonel Bogey* as the "You lose" music on a number of games. It is surprising how a relatively simple game can benefit by the addition of a few interesting noises — the player interest seems to increase by a factor of 10.

The otherwise very good manual supplied with the MZ-80K fails to mention that the sound capability of the machine goes very much further than the Music statement, as anyone who has played the Sharp version of Space Invaders will know.

It is a reasonably easy matter to produce a whole range of weird and wonderful noises and effects from a were-wolf to a wolf-whistle, from a telephone to a motorbike. With a galaxy of *glissandi* and space noises in between.

The secret lies in two monitor subroutines and two memory locations which between them control the way the sound is generated. It is easy to access these from Basic via the Poke and USR instructions.

The machine takes two numbers from one to 255, which are Poked into memory locations 4513 and 4514 and uses them to divide down the 2MHz clock which controls the CPU. Monitor subroutines are then used to start and stop the sound. These subroutines are called by USR(68) to start the sound and USR(71) to stop it.

This may seem a little complicated but it is simple when you are used to it and it operates very much faster than the Music command of Basic. Combining all this with For-Next loops means you can really begin to take control of the synthesiser and produce sounds that are limited only by your imagination.

The best way to become acquainted with this facility is to experiment. Try the following short routines.

```
10 FOR A = 255 TO 0 STEP -12
20 POKE 4514,1: POKE 4513,A
30 USR(68): NEXT A: MUSIC "R2"
40 FOR A = -255 TO 255 STEP 8
50 B = ABS(A): POKE 4514,1: POKE 4513,B
60 USR(68): NEXT A: USR(71)
```

That is how to produce a wolf-whistle. Try altering the values of A and see what happens to the sound. Note also in line 30 the use of Music "R2" in place of USR(71). Try switching them and note the difference.

To produce a photon torpedo or laser gun you could base your experiments on:

```
10 FOR T = 1 TO 4: FOR A = 1 TO 150
20 POKE 4514,A: USR(68): NEXT A
30 USR(71): NEXT T
```

That is even simpler than it looks. The For-Next loop of T determines how many times the gun is fired. The actual program for the effect will fit on one line. Again, try altering the value of A and see what happens. Replace the USR(71) with Music "R2" and note the effect on the gap between the shots.

Music "Rx" where x is a value from 0 to 9 is another way of stopping the sound but does have a different effect depending on the value of x.

A Trimphone-type telephone might find an application in a program:

```
10 FOR A = 1 TO 3: FOR B = 1 TO 2
20 POKE 4514,1: FOR C = 1 TO 15
30 POKE 4513,150: USR(68)
40 FOR D = 1 TO 6: NEXT D
50 POKE 4513,255: USR(68)
60 FOR E = 1 TO 6: NEXT E
70 NEXT C: MUSIC "R3"
80 NEXT B: MUSIC "R7": NEXT A
```

This routine will also supply the ringing tone and the engaged tone by altering the values in lines 20, 30 and 50. This is a good routine with which to experiment by altering all the values and noting the effect. Note once again the use of Music "Rx" in place of USR(71); exchange them just to see what happens.

The programs for different sounds are quite similar to each other which makes is possible, for instance, to have only one subroutine in a program but obtain several sound effects from it merely by altering the values of the variables when the routine is called.

I have written a short program to demonstrate some of the sounds available from the MZ-80K and to show how to go about achieving them. Anyone who cares to send me one of their programs and an SAE for the return of the cassette is welcome to a copy. The address to write to is 95 Bowring Park Avenue, Liverpool L16.

If you have ever played the game of Star Chess you will be familiar with the "Game won" noise; this produces something similar:

```
5 FOR T = 1 TO 5
10 FOR A = 10 TO 1 STEP -1
20 POKE 4514,A
30 FOR B = 0 TO 255 STEP A
40 POKE 4513,B: USR(68)
50 NEXT B,A: USR(71)
60 NEXT T
```

■

(continued from previous page)

maintain a duration table pointer at machine-code level.

The problem our program has is that it can cope with only one set of sound-effects data. If in a game we wanted several different effects we would have the slow job of Poking each sound sequence into memory before we could hear it.

It would be much better to have a situation in which we could store all the sounds in advance and then tell the sound-maker where to find the set we wanted at any time.

The machine code knows where to look for the sequence of notes we want it to play because it is given the address at which this data starts. At the moment, the routine looks at address 32489. Suppose we had several blocks of data, as in figure 4, each stacked above the other. All we need to do is feed address "x" to the routine to hear the flying saucer, address "y" to hear the laser gun.

The complication is that the machine-code routine cannot accept a whole address. The address must be split into two bytes, called the most- and the least-significant bytes respectively — MSB and LSB.

The mathematics to calculate these values for any address is not difficult, and the computer will crunch the numbers for you. If N is the address, then

MSB = INT (N/256)
LSB = N − (MSB * 256)

Now we have our stacked sound effects, we have the addresses at which they start converting into an MSB and LSB. All we need to know now is how to convey this information to the sound-making routine.

This, at least, is simple enough. You need the LSB in the second byte of the routine, and the MSB in the third byte. You can amend the current Basic program by inserting the following just before making the USR call:

POKE 32459, LSB: POKE 32460, MSB :

You can alter the MSB and LSB as much as you like but they must always go into the third and second bytes of the machine-code routine. You can even try putting some ROM addresses there. The computer will then make music from its own ROMs until it hits a zero.

Assembly-language programmers will have realised that a major weakness of the routine we are using to generate the sound is that while the sound is playing, the CPU bothers about nothing else. Most of the time it sits, gently contemplating its Nor gates, waiting for the B register to go zero, as in lines like

LOOP DJNZ LOOP

This means that a graphics-orientated game would not receive any screen updates while the sound was playing. A solution might be to replace the simple idea of consuming time by decrementing a register. Set the port to 1 or 2 and then check and update the board. You could control the time, and hence the frequency of the sound by controlling the number of T-states your screen check involved.

Those intrigued by the use of assembly language and who want to know more about the subject could do worse than buy Barden's book. Despite its relentlessly cheery approach, it is as gentle an introduction to assembly language as any I have read and a good deal cheaper than most. ■

Listing 1.

```
        LD      C,0; this will be duration.
NOTE    LD      B,0; this will be frequency.
        LD      A,1
        OUT     (OFFH,A; top half of wave
LOOP1   DJNZ    LOOP1; hold the wave for this long.
        LD      B,0;  this is the frequency again.
        LD      A,2
        OUT     (OFFH),A; bottom half of wave
LOOP2   DJNZ    LOOP 2 ; hold the wave for this long.
        DEC     C; duration counter
        JR      NZ,NOTE; go back if duration not finished.
        RET             ; go back to BASIC
```

Listing 2.

```
10 POKE 16562,126:POKE 16561,178:CLEAR 50:CLS:REM
      This is a memory size fix suggested by John Newgas.
20 DATA 14,0,6,0,62,1,211,255,16,254,6,0,60,211,255,16,254,13,32,
238,201
30 A=32437: REM start address for machine code.
40 MA=INT ( A/256) : LA=A- (MA * 256):REM usr pointers.
50 POKE 16527, MA: POKE 16526, LA: sets up USR call
60 FOR X = A TO A + 20: READ Z: POKE X,Z: NEXT:REM
      this line pokes in the m/c routine.
70 INPUT "DURATION"; D
80 INPUT "FREQUENCY"; F
90 POKE A + 1, D: POKE A+3,F:POKE A+11,F: REM data to routine
100 FOR GOES = 1 TO 10
110 S = USR(0)
120 NEXT
130 GOTO 70
```

Listing 3.

```
10......lines 10 to 60 are as for LISTING  TWO
65 FOR GOES = 1 TO 3
70 CLS:POKE A+1,50:POKE A + 11,110 : POKE A + 3,110
80 FOR L=1 TO 7 : PRINT:NEXT
90 A$= "WE HAVE VISUAL SIGHTING OF INTRUDER ABOVE BASE ZERO-4"
100 FOR L = 1 TO LEN (A$)
110 L$= MID$(A$,L,1)
120 PRINT L$;
130 F = USR(0)
140 NEXT
150 FOR DELAY = 1 TO 100 : NEXT DELAY, GOES
160 CLS
170 Y = 3: POKE A+1, 20: POKE A+3, 50; POKE A + 11,50
180 FOR X = 41 TO 43: SET (X,Y);  NEXT
190 FOR X = 40 TO 44 STEP 2: SET(X,Y+1):NEXT
200 FOR X = 38 TO 46: SET (X,Y+2): NEXT
210 FOR X = 910 TO 925: PRINT@X, CHRS (191);:NEXT
220 X = 44
230 FOR Z = Y + 3 TO 47
240 SET (X,Z)
250 F = USR(0)
260 RESET(X,Z)
270 IF POINT (X,Z+1) THEN 280 ELSE NEXT
280 GOTO 230
```

Listing 4.

```
65 A = 32437: POKE 16526,181:POKE 16527,126
66 REM in the data lines, frequency is first, duration second.
70 DATA 66,50,70,45,80,40,90,35,100,28,110,23,120,19,138,15,0,0
80 N= 1
90 READ F(N),D(N)
100 N= N+1
110 IF≠F(N-1) = 0 THEN 90:REM THE SIGN IS 'NOT EQUAL TO'
120 INPUT " TEMPO. . .F OR S ":AS: IF AS = "S" THEN M = 3 ELSE M= 1
130 FOR T = 1 TO N - 2
140 R (T) = D(T) * M
150 POKE A + 1,R(T): POKE A+3,F(T):POKE A + 11,F(T)
160 S = USR (0)
170 NEXT
180 INPUT " WANT TO HEAR IT AGAIN"; AS: IF AS = "YES" THEN  120
```

Listing 5.

```
130 T = RND(8)
140 R(T) = D(T) *  M:R1(T)= F(T) * 1.25
150 POKE A + 1,R(T): POKE A+3,R1(T): POKE A + 11,R1(T)
160 S= USR(0)
170 F$ = INKEY$: IF F$ = "" THEN 130
180 REM ** W.A MOZART IS ALIVE AND WELL AND LIVING IN THE Z-80
190 REM    HIT ANY KEY TO STOP MUSIC
```

Listing 6.

```
10  DEFINT A-Z
20  A = 32437
30  POKE 16527, 126: POKE 16526,181
40  R = PEEK(12488) : REM THIS IS NUMERAL KEYS 0 - 7
50  IF R = 0 THEN 40
60  IF R = 1 THEN F = 33: D = 50: GOTO 140
70  IF R = 2 THEN F = 35: D = 45: GOTO 140
80  IF R = 4 THEN F = 40: D = 40: GOTO 140
90  IF R = 8 THEN F = 45: D = 35: GOTO 140
100 IF R = 16 THEN F = 50: D= 28: GOTO 140
110 IF R = 32 THEN F = 55: D= 23: GOTO 140
120 IF R = 64 THEN F = 60: D= 19: GOTO 140
130 IF R = 128 THEN F= 69: D = 15
140 POKE A+1,D:F = F + F : POKE A+3,F: POKE A+11,F
150 S= USR(0) : F=0: D=0 : GOTO 40
```

Listing 7.

```
          LD    HL, NOTES; put address of note table in HL
          LD     E, DURTIN; duration will be POKEd in from BASIC
          LD     D, (HL); D hold current frequency.
KPON      LD     C,E    ; give the duration to C for decrements
SING      LD     B,D    ; give pitch to B for decrements
          LD     A,1
          OUT    (225),A
LOOP1     DJNZ   LOOP1 ; wait till first half of wave done.
          LD     B,D   ; restore the frequency counter.
          LD     A,2
          OUT    (255),A
LOOP2     DJNZ   LOOP2; wait for second half of wave
          DEC    C
          JR     NZ,SING; keep going on same note.
          INC    HL    ; move on the note pointer
          LD     D,(HL) ; get the data and
          XOR    A
          CP     D      ; see if it's zero
          RET    Z      ; back to BASIC if it is.
          JR    KPON    ; play next note if it's not.
```

Listing 8.

```
10   DEFINT A-Z
20   REM  first set of DATA is machine code instruction list
30   DATA 33, 242,126,30,00,86,75,66,62,1,211,255,16,254,66,62,
          2,211,255,16,254,13,32,239,35,86,175,186,200,24,231
40   REM  next lot of DATA is sound effect
50   DATA 80,64,48,32,16,9,8,7,6,5,4,3,2,1,2,3,4,5,6,7,8,9,16,32,
          64,80,96,112,128,144,0
60   FOR X = 32458 TO X+30
70   READ D: POKE X,D
80   NEXT
90   READ D:POKE X,D
100  X = X + 1
110  IF D ≠ 0 THEN 90
120  REM an OD ERROR crash means no zero at end.
130  A = 32458 : AM = A/256 : AL = A - (AM * 256)
140  INPUT "WHAT SPEED DO YOU WANT..BEST TRIES BETWEEN 5 & 15";S
150  POKE 32471,S
160  POKE 16527,AM : POKE 16526,AL
170  FOR GO = 1 TO 10
180  F = USR (0)
190  NEXT GO
200  GOTO 140
```

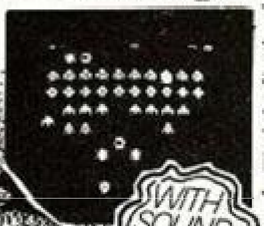
Listing 9.

```
49 REM this is DATA for flying saucer.  Suggest duration of 10.
50 DATA 80,73,72,72,68,67,66,65,64,63,62,61,60,59,58,57,56,55,54,53,
       52,51,50,49,0
       49 REM this is the gallop of the steel horse if duration 6 or
is
the mystical engine if duration is 10
50 DATA 48,64,80,96,112,128,144,160,176,192,208,209,210,211,212,
       213,214,215,216,221,173,189,205,221,237,254,14,94,158,239,
       15,159,225,0
```

The Essential Software Company

THIS IS NOT A PRACTICE DRILL! EARTH IS BEING INVADED ON YOUR TRS80 & VIDEO GENIE

Galaxy Invasion



The newest and most exciting invaders type game yet! Cruel and crafty aliens attack Earth. You are the sole defender. As you fire your laser at the aliens they swoop down and bomb you. Exciting use of graphics! Must be seen.

WITH SOUND

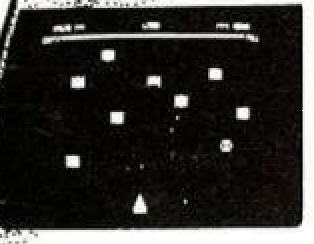
TRS 80 Level I & II 16K Tape

Video Genie 16K Tape

SuperNOVA



ORDER NOW

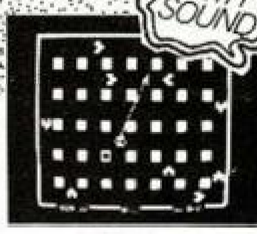


Now the amazing ASTEROIDS arcade game for your TRS 80! Your ship is floating in the middle of an asteroid belt! Your only escape is to destroy them and the crafty alien spacecraft! Blast them with your laser, thrust, rotate or hit hyperspace to survive!

TRS 80 Levels I & II 16K Tape

Video Genie 16K Tape

Attack Force



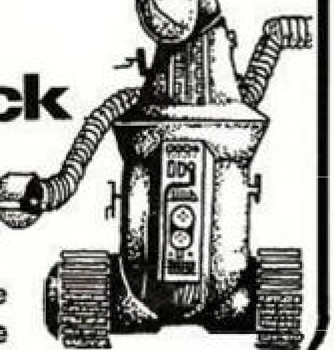
WITH SOUND

Dodge the alien Ramships and fire missiles to destroy them before they get you. The alien Flagship uses his deadly laser bolt to transform a Ramship into another Flagship or into your ship's double. Look out! Destroy your double and you could destroy yourself.

TRS 80 Level I & II 16K Tape

Video Genie EG3003 16K Tape

Robot Attack



THE GAME THAT TALKS

TRS 80 Levels I & II 16 K Tape

Video Genie 16 K Tape

The Newest and Most Astounding Arcade Game that TALKS has just Reached Planet Earth. You can't help yourself. You have to stop them at all cost. Don't let up. Written especially for high quality graphics you'll simply be dazed and excited by the action.

Cosmic Fighter

Your fighter appears below a convoy of Aliens! If you destroy them another set appears who seem to be slightly cleverer than before! Soon your space station nears but before you can dock the station comes under attack! Survival is up to you! The excitement is just beginning!!

TRS 80 Levels I & II 16K Tape

Video Genie 16K Tape

Gobble MAN



Watch out behind you! As you hurry through the maze collecting your energy modules you score points. But don't let the Gobblemen catch you. If you are crafty, sneek up behind them and neutralise them to gain extra points. Just keep a watch. When they attack you they come in fast. Just don't lose your nerve.

TRS 80 Levels I & II 16 K Tape

Video Genie 16 K Tape

3D Adventures

for TRS 80 LEVEL II 16K

3-D means that as you wander through the mazes and buildings, full screen graphic display constantly shows your position in a perspective format as though you were actually there! This "rat's eye" view adds an entirely new dimension to adventure.

English language commands can be entered at any time to manipulate your environment. The command sets are extensive and sophisticated. Dozens of objects are scattered throughout the mazes and buildings. You can pick them up, burn them, throw them, etc. You may need the sword to fight off an ugly little man. Or a steel rod to hold apart crushing walls. Deathmaze 5000 and Labyrinth allow the traditional one and two word commands. Asylum incorporates our Advanced Language Interpreter (ALI), which allows full sentence input.

Deathmaze and Labyrinth consist of over 550 locations! Asylum tops 1200 locations!

All Tapes £10 ea

ASYLUM Tape £12

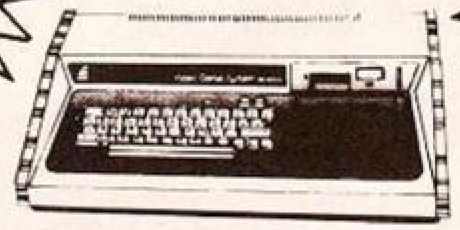
THE ESSENTIAL SOFTWARE COMPANY
(Viscounti Ltd.) 01-837 3154
47 Brunswick Centre, London WC1N 1AF

I have a.................................... microcomputer.
☐ Please send me your software catalogue. I enclose a stamped self addressed envelope.
☐ Please send me....................................
I enclose a cheque/postal order for £......
(plus 50p post & packing)

Signature ....
Name ....
Address ....
Postcode ....
My ACCESS No is ....

Video Genie

13K ROM version available



16K £275 + VAT

- ★ Latest version with vu-meter & extra keys
- ★ 16K RAM, 12K Microsoft Basic
- ★ Includes software, manuals and leads
- ★ All our machines are fitted with lower case characters as standard
- ★ 32K Ram memory version of Video Genie £329 + VAT

£329 + VAT 32K RAM

Optional Extras

- The unique 1K 'BIONIC' Rom £29.50 + VAT
- Sound unit £15 + VAT
- Double joysticks with software £28 + VAT
- Lower case characters with '£' sign. £20 + VAT

We stock a wide range of Genie accessories for the businessman and the hobbyist, Word processing, stock control, ledgers systems plus games. Refer to catalogue

BIONIC ROM

The Rom enhancement for your Video Genie will add the following immediately accessible features to your machine:

- Lower case driver
- Keyboard debounce
- Shift lock
- Repeat key routine
- Screen printout
- Cursor select
- Keyboard lock
- Bleeping keyboard etc.

Available only from us.

Kit £25 + VAT fitting £4.50

ATARI®

400 & 800
Computer Systems

ATARI 400 £295 + vat
ATARI 800 £543 + vat

SAVE!



EPSON MX-80

Letter quality matrix printer, has full software control of 40, 88, 66 or 132 columns 80cps bidirectional, disposable print head. There's lots of printers to choose from, but once you have compared you will find it hard to pass up to the MX-80. £ call

Video Genie interface £35 + VAT
Other interfaces available.



ACORN ATOM

8K ROM, 2K RAM BUILT £149
12K ROM, 12K RAM BUILT £229
Power supply £10.20
Colour moduator £23
all plus vat

Available shortly

- The BBC Microsoft basic upgrade
- 'Plug in' disc drive unit

Q-Tek Systems Ltd.

2 Daltry Close Old Town Stevenage Herts
Tel: (0438) 65385

Send £1 for latest catalogue (refundable). Please add vat to all items. Orders under £50 add 60p p + p otherwise carriage at cost

Phone for availability of **VIC 20** *Computer*

ZX-81 strings unravel knotty storage problems

Strings and their allied functions constitute a valuable armoury for users of any machine who wish to write elegant Basic. For the ZX-81, with its relatively low storage threshold, these powerful techniques are vital. Graham Thomson explains.

A ZX-81 WITH 16K RAM is a useful little machine. Among the most attractive facilities of its Basic is the string array used in conjunction with the functions Code and CHR$ and the slicing operator, To. Some powerful programming techniques can be implemented with these tools — and you do not need to have recourse to assembly language.

Most Basics will have equivalent functions, so the techniques I shall discuss should be generally applicable. The particular array used is that declared as, for example:

```
10 DIM S$(2000)
```

Unlike the undeclared string, the S$ defined has the very useful property that its length

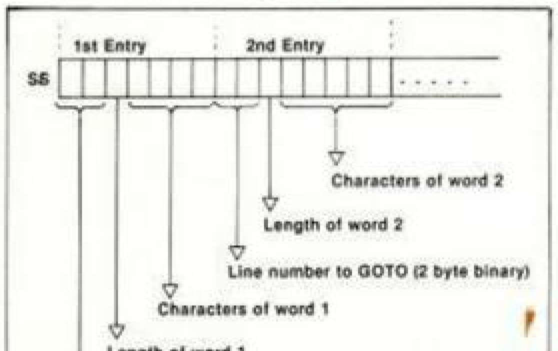


Figure 2.

never varies. We can, therefore, access the individual bytes at random using, e.g.,

```
S$(I) or S$(I TO J)
```

for both reference and assignment. In particular, we can maintain a simple numeric variable, say, NFB, which points to the "next free byte" in S$. Obviously, we set this to one initially and as we add to S$, we can increment NFB. For example:

```
200 LET S$(NFB TO NFB-1+LEN A$)=A$
210 LET NFB=NFB+LEN A$
```

adds A$ into S$ at the next free position, and adjusts the next free-byte pointer. It is true that you can achieve the same effect with an undeclared S$:

```
300 LET S$=S$+A$
```

but the easiest way, as so often, has disadvantages.

Firstly, in many Basics line 300 will be executed by constructing a shadow copy of S$,

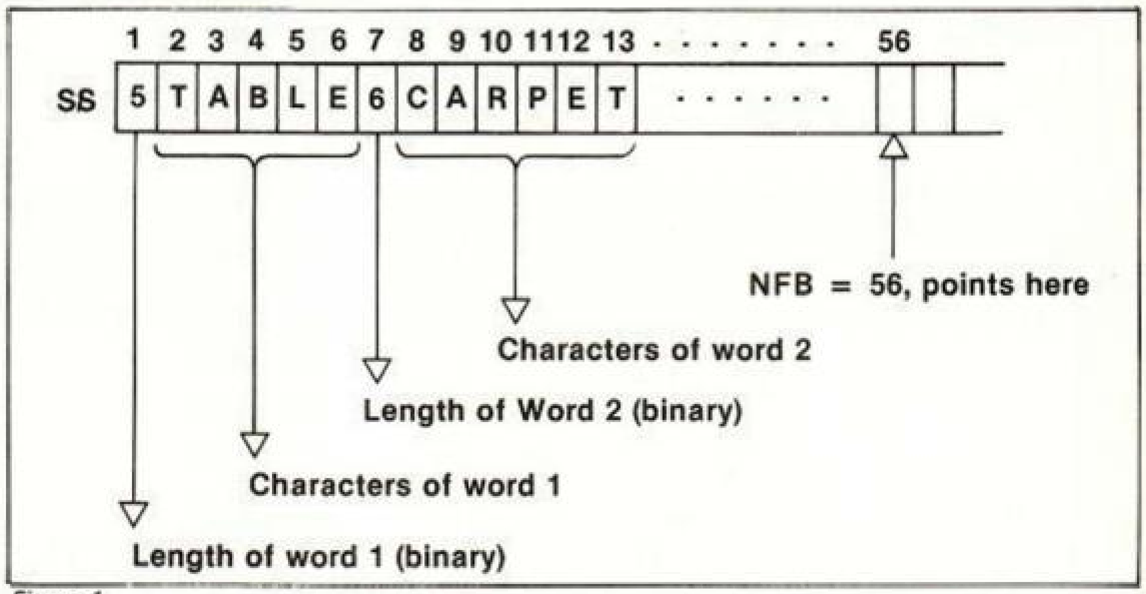


Figure 1.

so that for a time there are two copies occupying precious space in your RAM. That is acceptable if S$ is short, but what about an S$ that is 1,000 or 2,000 characters? Secondly, the easy way does not work with the more complex structures I shall outline. In the following examples, assume S$ declared as before and NFB initialised to one.

Now what might we want to store in this string-and-pointer structure? An obvious starting example is a list of words — such as you might want in that good old classic Hangman. A very simple piece of code to construct such a list is:

```
90 LET NW=0
100 INPUT A$
110 IF A$="END" THEN GOTO 170
120 LET S$(NFB)=CHR$ LEN A$
130 LET S$(NFB+1 TO NFB+LEN
    A$)=A$
140 LET NFB=NFB+1+ LEN A$
150 LET NW=NW+1
160 GOTO 100
170 . . .
```

This builds a list of words in S$. Each word is preceded by one byte containing, as an eight-bit binary number, the number of characters in the word. The variable NW counts the number of words.

Obviously, one would like to refine the code to ensure that no attempt is made to extend the list beyond the end of S$. I find it helps to draw simple diagrams of the structures being built in strings — figure 1 shows how the simple list of words appears in S$.

To pick words from the list — choosing, say, the next word in Hangman — we can count along using the length bytes to skip over words. For example:

```
400 LET N= INT (RND *NW+1)
410 LET P=1
420 LET N=N-1
430 IF N=0 THEN GOTO 460
440 LET P=P+1+ CODE S$(P)
450 GOTO 420
460 LET A$=S$(P+1 TO P+ CODE S$(P))
470 . . .
```

The code sets A$ to the Nth word in the list, where N is created as a random number between 1 and NW, the number of words. Note how the variable P hops along S$ always pointing to one of the length bytes. This type of sequential search can take several seconds on a ZX-81 if you have a long list of, say, 200 words.

The general principle which we have seen is that we can form in a single string, a set of list entries which constitute a simple list. This particular list entry had two components or "fields":

- Field 1 was one byte long and contained the length of field 2.
- Field 2 was of variable length and contained a word.

We can expand on this theme and add further fields to form more complex list entries. For example, we can add a two-byte field containing a line number. This gives us the type of list which can be employed to control a program using single-word, English commands. Figure 2 shows this type of list entry. Lists of this type, where there are several fields, are commonly called "tables".

The code to build this kind of control table could be:

```
30 PRINT "WORD?"
40 INPUT W$
50 IF W$="END" THEN GOTO 150
60 PRINT "LINE NO.?"
70 INPUT L
80 LET A=INT (L/256)
90 LET S$(NFB)= CHR$ A
100 LET S$(NFB+1)= CHR$ (L-256*A)
110 LET S$(NFB+2)= CHR$ LEN W$
120 LET S$(NFB+3 TO NFB+2+ LEN
    W$)=W$
130 LET NFB=NFB+3+ LEN W$
140 GOTO 30
150 . . .
```

(continued on next page)

(continued from previous page)

Lines 80 to 100 convert the line number, L, to a 16-bit binary number stored in the first two bytes of table entry. The code used to control such a table might be:

```
200 PRINT "WHAT NEXT?"
210 INPUT A$
220 LET P = 1
230 LET L = LEN A$
240 IF L<> CODE S$(P + 2) THEN GOTO 260
250 IF S$(P + 3 TO P + 2 + L) = A$ THEN GOTO
    300
260 LET P = P + 3 + CODE S$(P + 2)
270 IF P<NFB THEN GOTO 240
280 PRINT "I DO NOT UNDERSTAND"
290 GOTO 200
300 LET X = 256* CODE S$(P) + CODE
    S$(P + 1)
310 GOTO X
```

Obviously, the line number extracted from the entry at line 300 would start the code to process whatever word was typed. Building control tables and lists can save space in your RAM if you are writing complex programs.

Other examples of uses of lists and tables are in storing the co-ordinates of plot points to

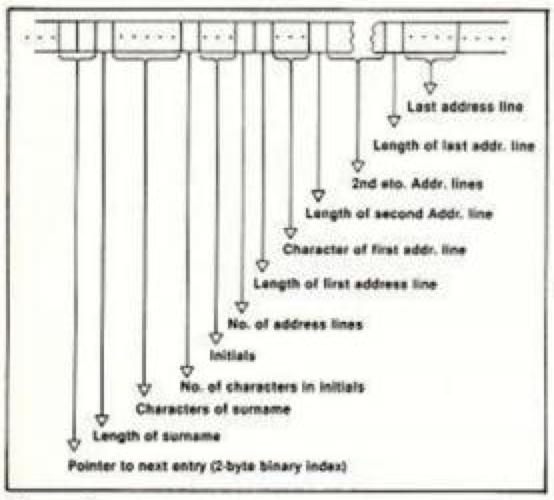


Figure 3.

draw diagrams or sections of diagrams. I have used this to store the plot points to draw the 10 separate stages of the Hangman picture. The list entry consisted of a one-byte count of plot points followed by two bytes for each point — one X co-ordinate, one Y co-ordinate. There were, of course, 10 such entries.

In a similar example, I wanted to draw large characters on the TV screen, specifically the digits 0 to 9 and the characters "+", "—", and "=". The characters were drawn using a four-by-seven grid of plot points, and each table entry included a count of plot points followed by their co-ordinates.

In fact, these were co-ordinates within the four-by-seven grid, so that the program could draw the large character anywhere on the screen by adding the full screen co-ordinates of the bottom-left corner of the character grid to the plot-point co-ordinates from the table.

It can be useful to keep a separate array of pointers to the table or list entries. It means you can construct a numeric array PTRS such that PTRS(I) gives the index into S$ of the Ith entry in the list or table. Naturally, if your list entries are all the same fixed length, you can calculate directly the index to the Ith entry.

In ZX-81 Basic such a pointer array requires five bytes per element. If you were particularly short on RAM, you might use another string, P$, with pairs of bytes holding a 16-bit binary numbers as indices into S$.

So the index of the Ith table entry is S$ would be

```
256* CODE P$(2*I) + CODE P$(2*I*1)
```

This is one of the situations where you need to consider whether or not the space saved by using a complicated string-based pointer system is wasted because of the extra program code needed to use it.

An interesting and often useful variation of the list or table concept is the chain or linked list. Here, one of the fields — usually the first — in each entry is, in fact, a pointer to the next entry. A simple example is a linked list of words.

The entries look just like figure 2, except that instead of a line number, the first two bytes of each entry hold the index to the next entry. A very obvious use of such a linked structure is in sorting lists. To sort the words in our example, there is no need to physically reorder the list entries — we just swap the pointers.

The use of such linked lists usually requires a separate variable to hold the index to the first entry of the chain. This variable is commonly called the head-of-chain pointer. Similarly, it is often useful to have another variable pointing to the last entry on the chain — the tail-of-chain pointer. The last entry on the chain usually has zero in the next-entry pointer.

A useful example of linked lists is a simple name and address program. The entries used have the format shown in figure 3. The following program code illustrates how this type of linked list can be manipulated.

It is a set of subroutines to request a complete name and address from the terminal. Add it into the string S$ at the next free byte, NFB, and adjust the links so that the list is maintained in alphabetical order of surname, irrespective of initials.

For convenience, the head-of-chain pointer is the first two bytes of the array S$. The subroutine at line 1600 prints all the names and addresses in alphabetical order by following the chain of pointers. On the first use of the subroutine at line 1000, NFB is 3, and the first two bytes of S$ are binary zero — space characters on ZX-81.

```
1000 REM SUBR. TO ADD AN ADDRESS
1010 CLS
1020 LET PNEW = NFB
1030 LET S$(NFB TO NFB + 1) = "  "
1040 LET NFB = NFB + 2
1050 PRINT "SURNAME?"
1060 GOSUB 1500
1070 PRINT "INITIALS?"
1080 GOSUB 1500
1090 PRINT "HOW MANY ADDRESS LINES?"
1100 INPUT N
1110 LET S$(NFB) = CHR$ N
1120 LET NFB = NFB + 1
1130 FOR I = 1 TO N
1140 PRINT "LINE ";I
1150 GOSUB 1500
1160 NEXT I
1170 REM NOW LINK IT IN CHAIN
1180 LET LP = 1
1190 LET P = 256* CODE S$(LP) + CODE
     S$(LP + 1)
1200 IF P = 0 THEN GOTO 1280
1210 LET A$ = S$(P + 2)
1220 IF A$>S$(PNEW + 2) THEN LET
     A$ = S$(PNEW + 2)
1230 LET L = CODE A$ + 2
1240 IF S$(PNEW + 3 TO PNEW + L)
     < = S$(P + 3 TO P + L) THEN GOTO 1270
1250 LET LP = P
1260 GOTO 1190
1270 LET S$(PNEW TO PNEW + 1) = S$(LP TO
     LP + 1)
1280 LET A = INT (PNEW/256)
1290 LET S$(LP TO LP + 1) = CHR$A +
     CHR$(PNEW—256*A)
1300 RETURN

1500 REM SUBR TO INPUT A LINE
1510 INPUT A$
1520 LET L = LEN A$
1530 LET S$(NFB) = CHR$ L
1540 LET S$(NFB + 1 TO NFB + L) = A$
1550 LET NFB = NFB + L + 1
1560 RETURN

1600 REM SUBR. PRINT ALL NAMES AND
     ADDRESSES
1610 LET P = 1
1620 LET P = 256* CODE S$(P) +
     CODE S$(P + 1)
1630 IF P = 0 THEN RETURN
1640 LET PT = P + 2
1650 GOSUB 1800
1660 LET B$ = A$
1670 GOSUB 1800
1680 SCROLL
1690 PRINT A$;" ";B$
1700 LET N = CODE S$(PT)
1710 LET PT = PT + 1
1720 FOR I = 1 TO N
1730 GOSUB 1800
1740 SCROLL
1750 PRINT A$
1760 NEXT I
1770 SCROLL
1780 PRINT
1790 GOTO 1620

1800 REM SUBR SET A$ = LINE AT PT
1810 LET L = CODE S$(PT)
1820 LET A$S$(PT + 1 TO PT + L)
1830 LET PT = PT + L + 1
1840 RETURN
```

Occasionally, it is even useful to create a double-linked chain. There, each entry contains two pointer fields; one to the next entry, one to the preceding entry. With such a chain you can process both forwards and backwards.

It is often the case that an entry in a chain or list consists of several fields, some of which are fixed in length, others are variable and need to be prefixed by a length byte. In general, it is convenient to group fixed-length fields at the beginning of the entry, followed by the variable-length fields.

Surprising as it may seen, the techniques used by writers of fundamental software — operating systems, compilers, interpreters — can be implemented in Basic. If you can think "pointers, lists, tables and chains" you can broaden the scope of your programming ability.

I have recently used all these techniques in a general-purpose Adventure program. It consists of the code to request descriptions of places — short description, long description, other places you reach by going north, south, east, west, objects and special words.

The place descriptions and linkages form a table, the object descriptions form chains. Each place entry has a head-of-chain pointer field from which will hang all of the objects dropped at that place.

Thus defining a new game for the children does not involve writing another program — just typing in the new place and object descriptions. ■

SILVERSOFT

personal computer software

ZX81/ 16K 'STARTREK'

16K STARTREK: GRAPHICS, 4-LEVELS OF PLAY, 8×8 GALAXY, STAR BASES, KLINGONS, ROMULANS, PHOTON TORPEDOS, ETC.

GAMES PACK 1: STARWARS, HAMURABI, GRAPHIC LANDER, MASTERMIND, MINEFIELD

Prices: £4.95 each cassette; £8.95 for both

Further games packs coming soon.
Large S.A.E. for details.
Mail order only.

SILVERSOFT
40 Empress Avenue
Ilford, Essex
01-518 0877

ZX81 Workstation...



. . . is a stylish and ergonomic plinth for the ZX81. It raises and tilts the TV to avoid eyestrain, holds the 16KRAM in place and hides the wiring and power supply. This very professional unit costs £15, a built-in power switch is £3, plus postage at £1.50, inc. VAT.

Peter Furlong Products, 125 Catford Hill, London SE6 4PR.
Callers by appointment, please. Tel 01690 7799. Visa, Access.



THE ZX81 POCKET BOOK

Also suitable for ZX80 with 8K ROM

- **ADVENTURE**
  - *City of Alzan*
  - *Create your own*
- **TUNNELS & TROLLS**
  BUCKET CATCHING
- **JAWS**
  BECOME AN ARTIST
- **PRO-AM GOLF**
  PUTTER
  ETCH-A-SKETCH
  FRUIT MACHINE
  With hold
  DIGITAL CLOCK
  DICE ROLLING
  plus 20 others

110 PAGE BOOK **5.95**

CASSETTE & BOOK **10.00**

★ require 16K RAM

PLUS Hints & tips on programming
Reference Sections

The ZX80 Pocket Book still available, prices as above.

ATOM BUSINESS

Suitable for the expanded Atom with floating point ROM

Each section contains:
- **Management Summary**
- **Operating Instructions**
- **Program notes**
- **Source listing**

SALES GRAPH
SALES RECORDS
NOMINAL LEDGER
QUEUEING SIMULATION
EXPENSE CLAIMS

ADDLIST
LABEL PRODUCTION
LEASE OR BUY DCF
METRIC CONVERSION
STANDARD DEVIATION
BUDGET FACTORING

110 Page Book **6.95**

Cassette & Book **15.57**

PHIPPS ASSOCIATES

3 DOWNS AVENUE EPSOM Surrey KT18 5HG
Tel: Epsom (03727) 21215 quoting your credit card reference. 24 Hour phone service. Air Mail delivery in Europe add 50p, air mail elsewhere add £1.50 per book. Prices shown above include UK postage and VAT on cassettes

ZX80

JRS SOFTWARE

19 WAYSIDE AVENUE, WORTHING, SUSSEX, BN13 3JU
TELEPHONE WORTHING 65691 (Evenings and Weekends only)

ZX81



ZX80 - PROGRAMMABLE MOVING DISPLAY
(4K ROM only)

Yes! This really is a **genuine** moving display, **not** another pause routine. If you want moving, flicker free displays [*and who doesn't*] then this is the program for you. The secret lies in the ZX80's ability to keep the display on your screen without the need to use all of the time available to it. Normally the ZX80 would be doing nothing during this spare time but the programmable moving display cleverly interupts to process your own instructions written in the simple but highly effective JRS numeric code. Great care has been taken so that the processing of your codes can always be interupted to return to the display routine at the precise microsecond that is required to ensure that your T.V. picture remains completely **rock-steady.**

Normally a true moving display on a ZX80 would take weeks to write and you would need to be an export at machine-code programming. Now, at last, this program offers you the ability to write your own true moving displays in under an hour with no machine-code experience required whatsoever.

Cassette with 1k, 2k versions and 3 example programs plus FULL documentation **£4.95**

** Awarded top place in ZX80 software review * in* ***'YOUR COMPUTER'*** *October 1981 issue*

NEW - An ESSENTIAL addition to your 1K RAM ZX81 (or ZX80 8K ROM)

TOOLKIT (written by PAUL HOLMES)

Provides the following additional facilities:-

Line renumber — you state starting number and increment value.
Search and replace — changes every occurence of a character as you require.
Free space —tells you how many free bytes you have left

SPECIAL GRAPHICS ROUTINES

Hyper graphics mode — graphics never seen on a ZX81 before.
Open — instantly sets up as many empty print lines as you require.
Fill — used in conjunction with OPEN fills your screen instantly with your specified character
Reverse — changes each character on your screen to its inverse video.

TAPE ROUTINE — provides a system **WAIT** condition until a signal is received in the cassette ear jack — many uses !

All these routines are written in machine code and together take up only 164 BYTES of your precious RAM - an incredible achievement!!

The price is incredible too! ONLY **£3.95** for cassette, including FULL instructions and example programs.

ALSO available 16K version ONLY **£4.95** which includes all the above PLUS:-
GOTO's and GOSUB's included in line renumber.
Search for and list every line containing specified character.

ZX81 2K programs SLALOM and BLACK HOLES now available as listings only £1.25 each, £2 for both.

OVERSEAS CUSTOMERS PLEASE NOTE Payment must be made in Sterling by International Money Order (available at your bank). Please add 50 pence to cover overseas postage.

ZX81

1K and 16K computer owners

Have you ever wondered if you are paying too much tax?

Check your payslip!

Your personal PAYE tax check is now available for year ended 5 April 1982.

Just enter the information from your payslip as instructed by the computer and your Tax Deductions, National Insurance Contributions, etc., will be checked and displayed for you to compare with your actual payslip. Remember an independent verification is the most satisfactory.

Program specifications:

16K
- Monthly or weekly
- Tax code change
- All rates of tax
- National Insurance tables A, B or C
- Not contracted out National Insurance
- Contracted out National Insurance

1K
- Monthly or weekly
- Basic rate tax
- National Insurance Table A
- Not contracted out National Insurance
- Contracted out National Insurance

and a tax code check is included

£2.95 per cassette inclusive from:

STROUD LITT AND CO.
85 Jamestown Road, London NW1 7DB

HILDERBAY LTD

Professional Programs for the Sinclair ZX81 + 16K Softare designed for people by computer professionals

THE HILDERBAY PAYROLL £25*
- 1 to 30 employees. Weekly, monthly, etc. No tables needed. Very easy to use.
- Payslips and summaries printed.
- All pay levels and numeric tax codes catered for. All National Insurance contributions (including contracted-out).
- Provision for pre- and post-tax changes for any reason (overtime, bonus, etc.).
- Last-minute changes no problems.
- Will also work backwards! (Complete gross and deductions from net pay.)
- Possible to go back to manual calculations at any time — you don't depend on your computer!
- Program maintained and updated by Hilderbay Ltd. Training not necessary, but can be arranged.
- Easily used by the non-specialist. Routine payroll quickly handled. Senior managers can process senior staff salaries confidentially.
- Check you own pay!
- Will handle a payroll of 500 employees(!) with 48K RAM pack!
- The Inland Revenue does not 'approve' computer programs (we did ask). The Hilderbay payroll satisfactorily deals with the 21 test cases of Inland Revenue systems unit note series 6 number 4 annex B.

THE HILDERBAY STOCK CONTROL PACKAGE £25
- Program I: up to 400 items in 16K; up to 2,000 in 48K. Numerical code, description, supplier code, type code, stock level, re-order level, unit price. Prints lists by supplier, by type, of understocked items, or all items.
- Program II: up to 2,500 items in 16K, 8,000 items in 48K. Numerical code and stock level only.
- Both programs are very fast! Locate, insert, delete an item from a list of 2,700 in well under two seconds!

THE GIBBONS PERSONAL BANKING SYSTEM £15*
- Do you sometimes wonder where all the money goes? Now you can find out!
- Keep your finances under control. Standing orders automatically allowed for.
- Do you wish your bank statements told you what each item was? Now you can print better statements than your bank can!

HILDERBAY CRITICAL PATH ANALYSIS (CPA) £15
ZX81 implementation of a powerful tool of modern management. Designed for practical use, but also suitable for learning about CPA.

HILDERBAY FINANCIAL PACK I £8
VAT: Given a mixture of bills with and without VAT at several rates, prints table of: price, VAT, total, VAT rate. Displays and prints running totals.
MORTGAGE: How much longer will your mortgage take to repay if you pay £25 less per month? (The answer may surprise you.) How much must you pay per month to repay in 15 years? How long to repay at £200 per month? How much of your latest payment went towards interest, and how much towards principal (the answer will shock you). Results given either for UK Building Society or (different) bank loan.
LOAN: Calculates one of the following in terms of the other three: principal, number of payments, instalment, interest.

HILDERBAY GAMES PACK I £8
GOLD: A rather difficult adventure. We guarantee that you can get there with the treasure — but we're not telling you how!
PICK A WORD: An unusual game of chance. Play against the computer, pick words from the list until one of you gets a winning set of letters.

CHESS £10
Plays excellent games. Several levels of play.

- All programs require 16K RAM pack. All prices include VAT, postage and packing. C.O.D. (cash on delivery — pay the postman) orders are accepted for a supplement of £2 per order. Urgent orders no problem.
- All programs are supported by Hilderbay Ltd — we won't abandon you to your fate. Cassettes are produced with great care and are guaranteed.
- Programs are designed for use by people who are not familiar with computers.
- All programs are supplied with full documentation. Manuals for programs marked (•) can be bought separately (£2, refundable on program purchase).
- Programs can be demonstrated (by appointment only). Further details available on request.
- 48K Memotechs demonstrated and sold.

Software also supplied for mainframe computers, Hewlett-Packard 9845 and 85, various microcomputers.
Hilderbay Ltd is not associated with any other company. Hilderbay software is now sold only by us and our authorised dealers.
Please telephone before visiting us.

HILDERBAY LTD (founded 1979)

8/10 Parkway, Regents Park, London NW1 7AA Telephone: 01-485 1059 Telex: 22870

School computers: some lessons to be learnt

Eric Deeson reviews the problems and progress of educational computing.

WE ARE NOW approaching the stage at which half of Britain's secondary schools have computer power available. Even the number of primary schools with computing facilities is no longer negligible.

A number of recent surveys, shown in table 1, have attempted to discover just how many computers there are in schools, but their results have been overtaken by the advent of Sinclair's ZX-81. Both the Commodore Vic and the eagerly-awaited Acorn/BBC machine are likely to have a significant impact on educational computing.

The Government plans to subsidise the purchase of micros for schools. It will pay half the cost of either the Research Machines 380-Z or the Acorn/BBC machine for schools which so far have no computer. This project has been much criticised on the grounds that support for specific machines might be unwise. It is also feared that schools which have shown little interest until now will continue to be inactive.

The educational use of computers can be broken down into a number of categories:

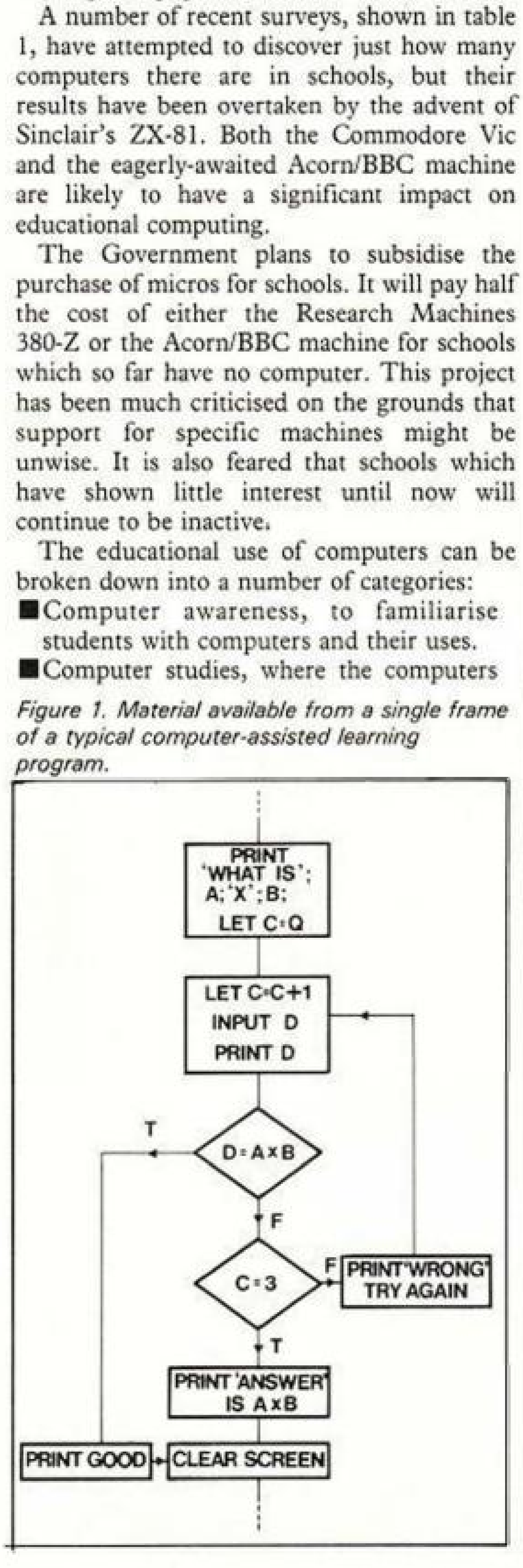
- Computer awareness, to familiarise students with computers and their uses.
- Computer studies, where the computers themselves are the object of interest.
- Computer-assisted learning, in which computers are used to teach other subjects.
- Administration, where a computer is used in helping to run the school.

	CET	CLS	MUSE
Commodore Pet	276	53	579
RML 380-Z	429	99	702
Tandy/Video Genie	53	9	125
ZX-80/81	—	9	87
Apple/ITT	115	25	252
Nascom	—	—	44
Acorn Atom	—	—	59

CET: Council for Educational Technology, 1980.
CLS: Cambridge Learning Systems, January 1981.
MUSE: Microcomputer Users in Secondary Education, July 1981.

Table 1. Recent surveys of computers in schools.

Figure 1. Material available from a single frame of a typical computer-assisted learning program.



Computer-assisted learning includes all aspects of computer use in which the machine plays a direct part in teaching the student. Computers can help teachers by taking over the dull, routine, repetitive tasks — like marking, calculating class orders, working out attendance records and scheduling pupils to resources — in fact, acting as a clerical assistant.

The most remarkable commercial system of this kind is Plato, which has come to Britain from the U.S. Users of Plato terminals can log in what they want with their own code. The computer guides them through their chosen courses, managing the statistics of their progress, enabling assessment, supervising the use of books, audio-visual and laboratory work, and providing interactive learning programs.

Within a decade, small straightforward micro-based systems of this versatility will be able to handle courses held on EPROMs or bubble memories. Plato is essentially a development of the programmed-learning technology of the 1960s, and there is still very little that is useful. Material available from a typical single frame is shown in figure 1.

Individual teachers are spending a lot of time trying to mirror their own teaching technique — one of flexibility, sophistication, and explanation — to two computer programs. An incorrect answer should lead to real help; a correct input should be checked for understanding.

Programming for teaching calls for a combination of teaching expertise and programming ability — and a great deal of time. Rush leads to rubbish, and there's plenty of rubbish around. One of the reasons for the failure of many mechanical teaching aids in the 1970s was the general inadequacy of the software.

To many people, computer-assisted learning is viewed as just one of the many resources available for a class. It is viewed in conjunction with the chalk board, a projector, maps and hand-outs, and specialist laboratory equipment. Its applications include:

- computation in geography, mathematics and the sciences;
- development, testing and using models in the practical and social sciences, and in sports;
- retrieval of stored data;
- graphical presentation of data;
- educational game-playing and assimilations in social, general and science subjects;
- reinforcement and testing;
- control of laboratory experiments and data capture.

Software problems and teachers' inexperience remain major bugbears. Though most schools with a micro probably have a couple of real enthusiasts on the staff, the majority of serving teachers have no knowledge of computing and its potential. Those teachers who are expert in computing still have a full teaching timetable of their own, and no time for good software development. Ideas often have to come from the pupils themselves.

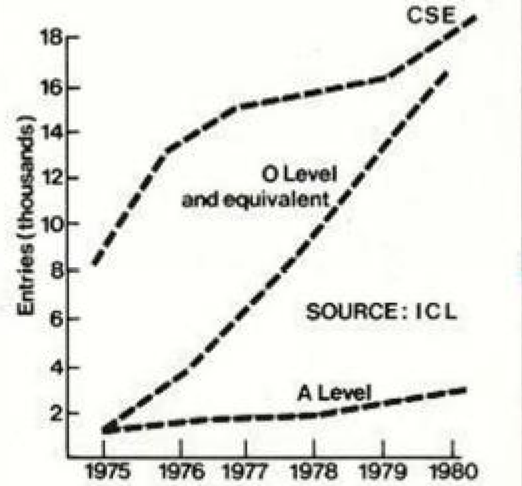


Figure 2. Number of entries for computer-studies papers in public examinations.

Though computer-assisted learning has not travelled far towards achieving its potential, it is probably the most important area of educational computing. Another important field at the moment is computer studies. There cannot be many people who would deny that schools should attempt to teach something of our new micro-world.

(continued on next page)

(continued from previous page)

That attempt is being made in several ways. Firstly, there are signs of changes in this direction in formal examination syllabuses, particularly in physics, maths and technology, where some knowledge of simple microelectronics and computer arithmetic is coming to be expected.

A number of examination boards now offer papers in computer studies — see figure 2. Opinions of the value of these examinations vary greatly, but even if they give little direct career benefit to the students, they do fulfil the essential task of helping teachers learn basic principles. The same applies to test editing in commerce departments, a related growing field.

Computer-awareness courses are potentially the most useful, giving a brief introduction to computers and their uses for all pupils. They are usually short courses, often run fairly informally as part of general or social studies.

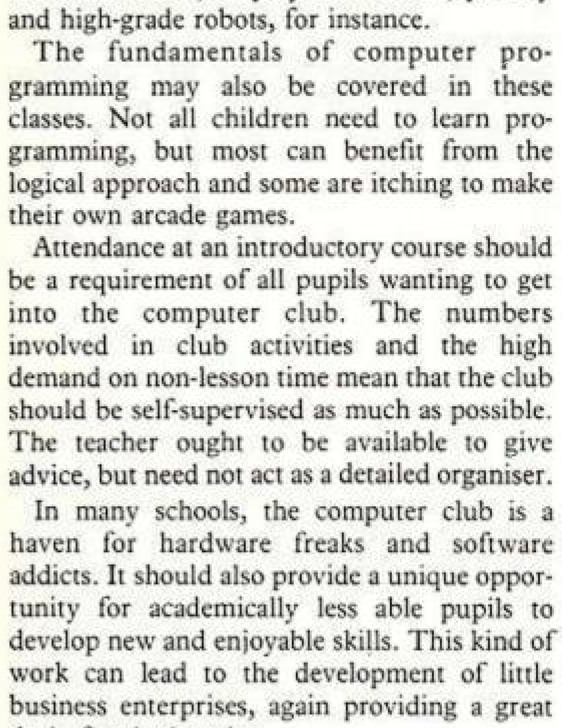
Their aim is to indicate in broad outline what computers are, what they do, and their possible effects on society. Topics which ought to be covered include the future electronic office, employment effects, privacy and high-grade robots, for instance.

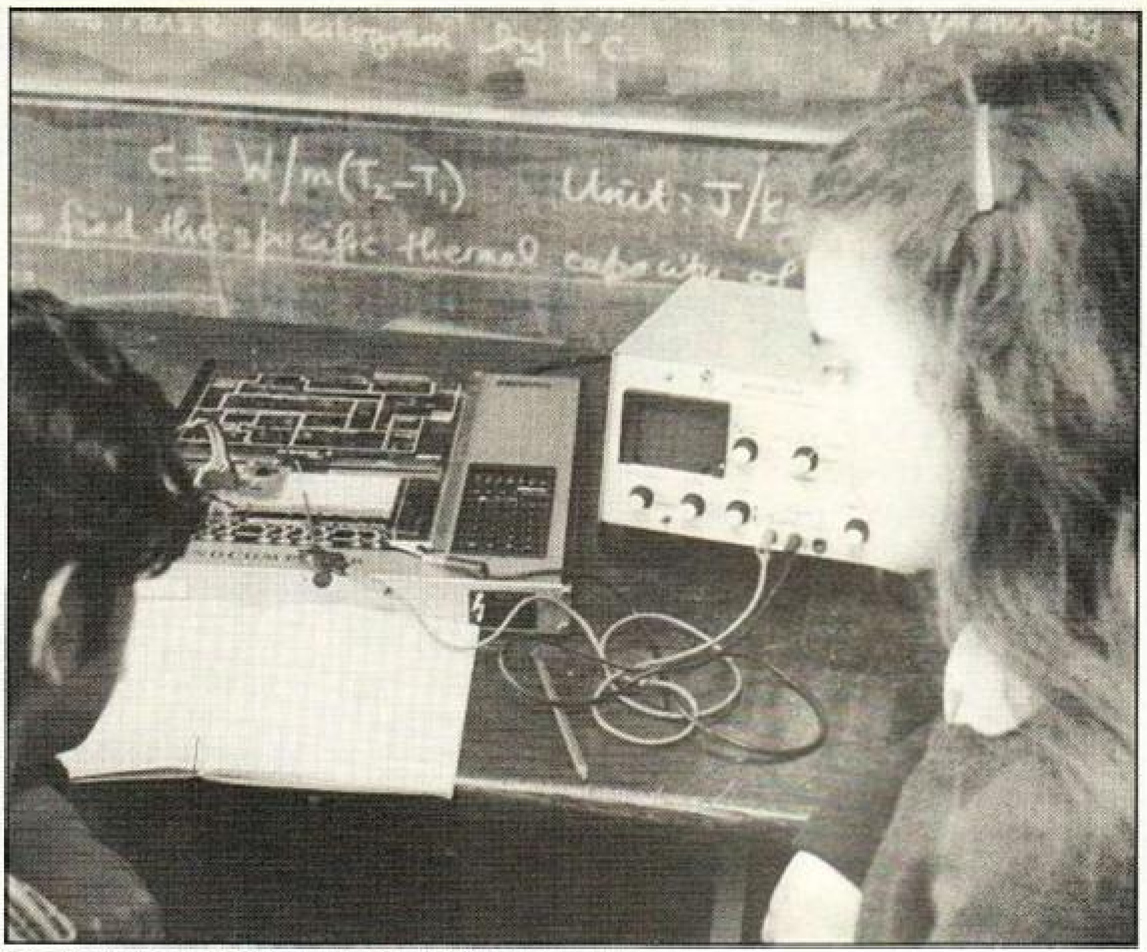
The fundamentals of computer programming may also be covered in these classes. Not all children need to learn programming, but most can benefit from the logical approach and some are itching to make their own arcade games.

Attendance at an introductory course should be a requirement of all pupils wanting to get into the computer club. The numbers involved in club activities and the high demand on non-lesson time mean that the club should be self-supervised as much as possible. The teacher ought to be available to give advice, but need not act as a detailed organiser.

In many schools, the computer club is a haven for hardware freaks and software addicts. It should also provide a unique opportunity for academically less able pupils to develop new and enjoyable skills. This kind of work can lead to the development of little business enterprises, again providing a great deal of real education.

Figure 3. The exercise book of the future? A small A4-sized computer could have tremendous potential in school and at home.





Practical microprocessor work with the Nanocomputer.

School administration by computer is beginning to take off. The direct beneficiaries are the staff who are relieved of much of their routine work.

There are now programs which make a fair attempt at timetabling — the deputy head's nightmare. Computer assistance in the library, resource centre, and general ofice can much improve working conditions. Some folk are even using Sinclairs to handle the tuck-shop accounts.

Data capture and process control are areas with enormous potential which are, as yet, hardly touched. Science laboratories and technology workshops are the sites for these developments in the first instance, but doubtless other areas will get the spin-off later. Computerised thermostats and the old favourite, model traffic-light control, are appearing at more and more school open days. Pressure from pupils will accelerate the process.

The pessimistic view for the future — one fairly often expressed — is that schools' computers will follow the 1960s' teaching machines into oblivion, leaving hardly a ripple behind. Of course, teaching is a pretty conservative profession which remains highly labour-intensive despite all the existing technological claims. Sceptics compare schools' computing to the audio-visual vogue of a decade ago. Certainly, a lot of money was wasted at that time, but now equilibrium has been reached at a much higher level of use than before.

At the other end of the spectrum are the visionaries who see schools reduced to social development or "play" centres. Learning would, they say, take place at home or in small units under the management of computers with little human intervention.

Who knows? Maybe that will come in a number of decades. Certainly there are some children I'd be glad to hand over to such a system, and many who would do well on it.

Between these extremes lie the more realistic possibilities. They will follow from progress towards large memory on cheap chips input pads, flat-screen output, plug-in EPROM language, programs and data.

Within five or 10 years we could have a flat A4-sized multi-purpose computer which would be capable of everything a 64K micro could do and more, and could communicate with the local databases and other machines. If the cost came down to around £30 at today's prices, it would have tremendous potential in schools as well as in the home — see figure 3.

Before that day dawns, we should all have developed a fair idea of how to use such a machine in the classroom. So let's get on with it.

Several national educational computing associations already exist. The largest and most active is MUSE — Microcomputer-Users in Secondary Education — whose membership gains access to a network of user, regional and local groups with many activities. The three-day annual meeting takes place in July. Details are available from Bob Trigger, MUSE, Freepost, Bromsgrove, B61 0JT for details.

There are far too many books on educational computing to give a comprehensive list here. One recent addition to the list is *Mindstorms* by Professor Seymour Papert, which could become very influential. It is published by Harvester Press at £9.95. *Run, computer, run* by Anthony Oettinger, published by Macmillan, may not be in print any longer — it first appeared in 1969 — but I still find a scan of it most salutory when my enthusiasm runs out of bounds.

Computing is exciting to many of our pupils, including the less able. It is richly rewarding to find a trouble-maker suddenly immersed in a program and thereby gain an interest in school work. Believe me, it does happen. ■

sinclair ZX81 PERSONAL COMPUTER





Sinclair ZX81 Personal Comp
the heart of a system that grows with you.



1980 saw a genuine breakthrough – the Sinclair ZX80, world's first complete personal computer for under £100. Not surprisingly, over 50,000 were sold.

In March 1981, the Sinclair lead increased dramatically. For just £69.95 the Sinclair ZX81 offers even more advanced facilities at an even lower price. Initially, even we were surprised by the demand – over 50,000 in the first 3 months!

Today, the Sinclair ZX81 is the heart of a computer system. You can add 16-times more memory with the ZX RAM pack. The ZX Printer offers an unbeatable combination of performance and price. And the ZX Software library is growing every day.

Lower price: higher capability
With the ZX81, it's still very simple to teach yourself computing, but the ZX81 packs even greater working capability than the ZX80.

It uses the same micro-processor, but incorporates a new, more powerful 8K BASIC ROM – the 'trained intelligence' of the computer. This chip works in decimals, handles logs and trig, allows you to plot graphs, and builds up animated displays.

And the ZX81 incorporates other operation refinements – the facility to load and save named programs on cassette, for example, *and* to drive the new ZX Printer.



New BASIC manual

Every ZX81 comes with a comprehensive, specially-written manual – a complete course in BASIC programming, from first principles to complex programs.

Kit: £49.95

Higher specification, lower price – how's it done?
Quite simply, by design. The ZX80 reduced the chips in a working computer from 40 or so, to 21. The ZX81 reduces the 21 to 4!

The secret lies in a totally new master chip. Designed by Sinclair and custom-built in Britain, this unique chip replaces 18 chips from the ZX80!

New, improved specification

- Z80A micro-processor – new faster version of the famous Z80 chip, widely recognised as the best ever made.
- Unique 'one-touch' key word entry: the ZX81 eliminates a great deal of tiresome typing. Key words (RUN, LIST, PRINT, etc.) have their own single-key entry.
- Unique syntax-check and report codes identify programming errors immediately.
- Full range of mathematical and scientific functions accurate to eight decimal places.
- Graph-drawing and animated-display facilities.
- Multi-dimensional string and numerical arrays.
- Up to 26 FOR/NEXT loops.
- Randomise function – useful for games as well as serious applications.
- Cassette LOAD and SAVE with named programs.
- 1K-byte RAM expandable to 16K bytes with Sinclair RAM pack.
- Able to drive the new Sinclair printer.
- Advanced 4-chip design: micro-processor, ROM, RAM, plus master chip – unique, custom-built chip replacing 18 ZX80 chips.

Built: £69.95

Kit or built – it's up to you!
You'll be surprised how easy the ZX81 kit is to build: just four chips to assemble (plus, of course the other discrete components) – a few hours' work with a fine-tipped soldering iron. And you may already have a suitable mains adaptor – 600 mA at 9 V DC nominal unregulated (supplied with built version).

Kit and built versions come complete with all leads to connect to your TV (colour or black and white) and cassette recorder.



uter-



Available now-
the ZX Printer
for only £49.95

Designed exclusively for use with the ZX81 (and ZX80 with 8K BASIC ROM), the printer offers full alphanumerics *and* highly sophisticated graphics.

A special feature is COPY, which prints out exactly what is on the whole TV screen without the need for further intructions.

At last you can have a hard copy of your program listings – particularly useful when writing or editing programs.

And of course you can print out your results for permanent records or sending to a friend.

Printing speed is 50 characters per second, with 32 characters per line and 9 lines per vertical inch.

The ZX Printer connects to the rear of your computer – using a stackable connector so you *can* plug in a RAM pack as well. A roll of paper (65 ft long x 4 in wide) is supplied, along with full instructions.

16K-byte RAM pack for massive add-on memory.

Designed as a complete module to fit your Sinclair ZX80 or ZX81, the RAM pack simply plugs into the existing expansion port at the rear of the computer to multiply your data/program storage by 16!

Use it for long and complex programs or as a personal database. Yet it costs as little as half the price of competitive additional memory.

With the RAM pack, you can also run some of the more sophisticated ZX Software – the Business & Household management systems for example.

How to order your ZX81

BY PHONE – Access, Barclaycard or Trustcard holders can call 01-200 0200 for personal attention 24 hours a day, every day.
BY FREEPOST – use the no-stamp-needed coupon below. You can pay by cheque, postal order, Access, Barclaycard or Trustcard.
EITHER WAY – please allow up to 28 days for delivery. And there's a 14-day money-back option. We want you to be satisfied beyond doubt – and we have no doubt that you will be.

To: Sinclair Research Ltd, FREEPOST 7, Cambridge, CB2 1YY. **Order**

Qty	Item	Code	Item price £	Total £
	Sinclair ZX81 Personal Computer kit(s). Price includes ZX81 BASIC manual, excludes mains adaptor.	12	49.95	
	Ready-assembled Sinclair ZX81 Personal Computer(s). Price includes ZX81 BASIC manual and mains adaptor.	11	69.95	
	Mains Adaptor(s) (600 mA at 9 V DC nominal unregulated).	10	8.95	
	16K-BYTE RAM pack.	18	49.95	
	Sinclair ZX Printer.	27	49.95	
	8K BASIC ROM to fit ZX80.	17	19.95	
	Post and Packing.			2.95

☐ Please tick if you require a VAT receipt TOTAL £

*I enclose a cheque/postal order payable to Sinclair Research Ltd, for £

*Please charge to my Access/Barclaycard/Trustcard account no.

*Please delete/complete as applicable.

Please print.

Name: Mr/Mrs/Miss

Address:

FREEPOST – no stamp needed. YOC12

sinclair ZX81

6 Kings Parade, Cambridge, Cambs., CB2 1SN. Tel: (0276) 66104 & 21282.

·' the ZX81 compares with other personal computers

SYSTEM IDENTIFICATION		ZX81	ZX80	ACORN ATOM	APPLE II PLUS	PET 2001	TRS 80 LEVEL I	TRS 80 LEVEL II
ROM		8K	4K	8K	8K	14K	4K	12K
GUIDE PRICE	Basic unit – inc. VAT	£70	£100	£175	£630	£435	£290	£375
	Unit plus 16K RAM (*12K RAM)	£120	£150	£285*	£630	£530	£360	£375
COMMANDS	LIST, LOAD, NEW, RUN, SAVE	●	●	●	●	●	●	●
STATEMENTS	PRINT, INPUT, LET, GOTO, GOSUB/RETURN, FOR/NEXT IF/THEN	●	●	●	●	●	●	●
	STEP	●		●	●	●	●	●
	TAB	●			●	●	●	●
ARITHMETIC	ABS, RND	●	●	●	●	●	●	●
FUNCTIONS	INT	●			●	●	●	●
	ATN, COS, EXP, LOG, SGN, SIN, SQR, TAN	●			●	●		●
	ARCSIN, ARCOS	●						
STRING	CHR$	●	●		●	●		●
FUNCTIONS	LEN	●		●	●	●		●
	ASC(CODE), STR$, VAL, INKEY$	●				●		●
NUMBERS	FLOATING PT $\pm10^{\pm38}$	●			●	●	●	●
	INTEGERS		●	●	●	●		●
NUMERIC	A-Z			●			●	
VARIABLES	AA-ZØ				●	●		●
	An-Zn, n = any alphanumeric string	●	●					
STRING	A$ & B$						●	
VARIABLES	A$ to Z$	●	●	●				
	An$ to Zn$ n = any alphanumeric character				●	●		●
NUMERIC	SINGLE DIMENSIONAL		●	●			●	
ARRAYS	MULTI DIMENSIONAL	●			●	●		●
DISPLAY	ROWS	24	24	16	24	25	16	16
	COLUMNS	32	32	32	40	40	64	64
	LOW RES GRAPHICS (<7000 pixels)	●	●	●	●	●	●	●
	HI RES GRAPHICS (>40000 pixels)			●	●			
SPECIAL	USR (CALL, LINK)	●	●	●	●	●		●
FEATURES	PEEK, POKE (OR EQUIV)	●	●	●	●	●		●

Sinclair software on cassette.



The unprecedented popularity of the ZX Series of Sinclair Personal Computers has generated a large volume of programs written by users.

Sinclair has undertaken to publish the most elegant of these on pre-recorded cassettes. Each program is carefully vetted for interest and quality, and then grouped with others to form single-subject cassettes.

Software currently available includes games, junior education, and business/household management systems. You'll receive a Sinclair ZX Software catalogue with your ZX81 – or see our separate advertisement in this magazine.

The ultimate course in ZX81 BASIC programming.



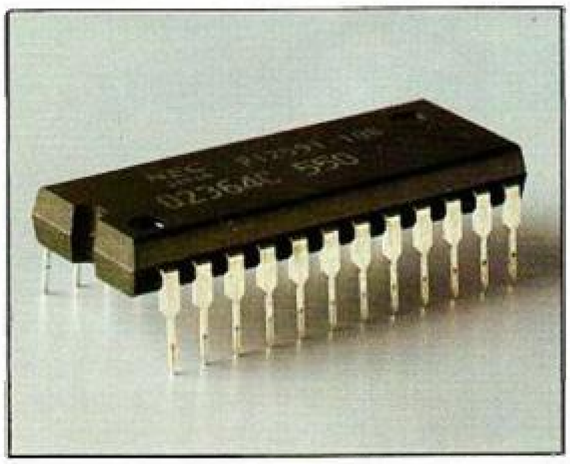
Some people prefer to learn their programming from books. For them, the ZX81 BASIC manual is ideal.

But many have expressed a preference to learn *on* the machine, *through* the machine. Hence the new cassette-based ZX81 Learning Lab.

The package comprises a 160-page manual and 8 cassettes. 20 programs, each demonstrating a particular aspect of ZX81 programming, are spread over 6 of the cassettes. The other two are blank practice cassettes.

Full details with your Sinclair ZX81.

If you own a Sinclair ZX80...



The new 8K BASIC ROM used in the Sinclair ZX81 is available to ZX80 owners as a drop-in replacement chip. (Complete with new keyboard template and operating manual.)

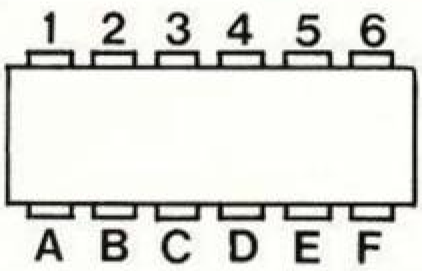
With the exception of animated graphics, all the advanced features of the ZX81 are now available on your ZX80 – including the ability to drive the Sinclair ZX Printer.

sinclair
ZX81

6 Kings Parade, Cambridge, Cambs., CB2 1SN.
Tel: (0276) 66104 & 21282.

VIC-20 CASSETTES

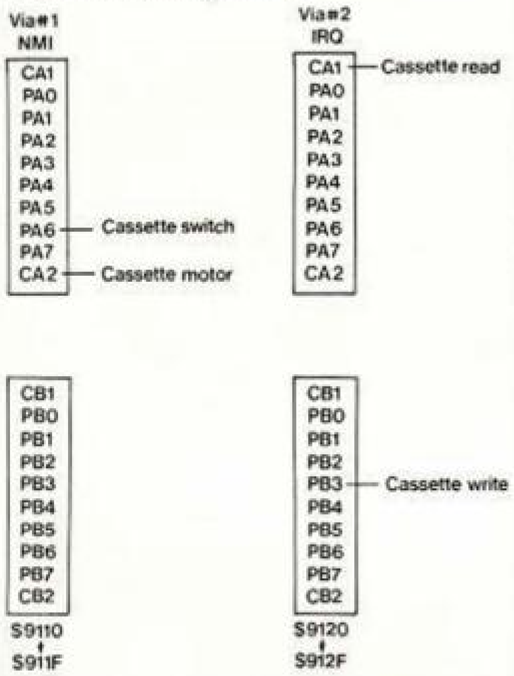
BY
NICK HAMPSHIRE



Pin#	FUNCTION
A-1	GND
B-2	+5V
C-3	Cassette motor
D-4	Cassette read
E-5	Cassette write
F-6	Cassette switch

Figure 1. Connector configuration.

Figure 2. VIA assignments.



Hardware

THE VIC HAS a single, external cassette unit which is used for program and data storage. This unit is connected to the Vic by six lines — write, read, motor, sense and two power lines, ground and +5V. The connections are shown in figure 1.

The cassette is controlled by I/O lines from the two VIA (versatile interface adaptor) chips, and you can see the source of each of the cassette-control lines from the VIAs in figure 2.

The cassette-motor power-supply lines are connected to the interface chips via a three-transistor driver which is used to boost the power and voltage — it allows the motor to be driven directly. The output to the motor is an unregulated +9V at a power rating of up to 500mA. The cassette-deck motor can be turned on and off by toggling the CA2 line on 6522 # 1.

POKE 37148, PEEK (37148) AND 241 OR 14 turns the motor on:

POKE 37148, PEEK (37148) OR 12 AND NOT 2 turns it off.

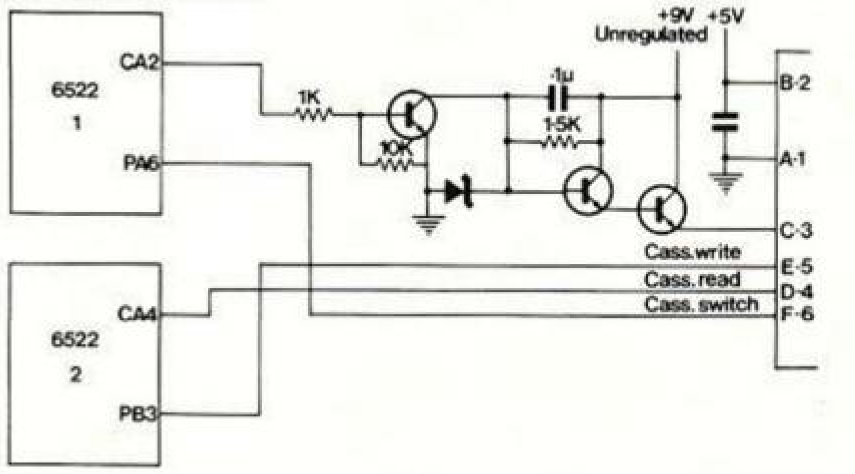
The sense-line input, line PA6 on VIA# 1, is connected to a switch on the cassette deck which senses when either the play, rewind or fast-forward buttons have been pressed. The switch is only required to sense whether or not you have pushed the play button during a read- or write-to-tape routine. This is done by a subroutine at $F8AB.

If either the rewind or fast-forward button is pressed accidentally instead of the play button, the system will be unable to tell the difference and will act as if the play button had been pressed. Because recording will start as soon as the play button has closed the sense switch, you must press the record button first in any record routine.

The cassette read line is connected to the CA1 line of VIA # 2 and the cassette write line to line PB3 of VIA # 2. During a read operation, the operating system uses the setting of the CA1 interrupt flag to detect transitions on the cassette-read line. The read and write lines are controlled entirely by the operating system — the only hardware required is signal-amplification and pulse-shaping circuitry.

These circuits are contained on a small, printed-circuit board within the cassette deck. Their function is to give correct voltage and current to the record head and to amplify the input from the read head. That gives a 5V square-wave output capable of producing an interrupt on the CA1 or CB1 lines.

Figure 3. Cassette interface circuit.



Cassette operation techniques

FOR NORMAL purposes the cassette deck is assigned the device number 1. The I/O number of the device currently in use is stored in location 186. This number, the logical file number, and the secondary address are used when saving or retrieving data files from the cassette deck.

The logical file number can be any number from 1 to 255 and is used to allow multiple files to be kept on the same device. It is of little use with cassette tape and is intended primarily for floppy-disc units. Usually the logical file number is the same as the device number and is stored in location 184.

Since it determines the operational mode of the cassette, the secondary address is important and the current one is stored in location 185. The normal default value is zero. If the secondary address is zero, the tape is Opened for a read operation. If it is set to one, it is opened for a write operation and if two, it is opened for a write, and an end-of-tape header is forced when the file is closed.

The Vic operating system is configured to allow two types of file to be stored on cassette: program files and data files. These names are however rather misleading since a program can be stored as a data file and data can be stored as a program file.

The difference between the two types is not in their application but in the way the contents of the machine's memory is recorded. Instead of program and data files, we must look on them as binary and ASCII files.

A binary file is usually used to store programs, since it is created by the operating system to store the contents of memory between a starting location and an end location. It is called a binary file because it stores on tape the binary value in each memory location within the assigned memory area.

Basic statements are stored in memory using tokens. The use of tokens means that Basic commands are not stored in the same manner as they are listed on the display or were entered from the keyboard. Instead, they are stored in memory in a partly-encoded form. Being partly encoded, a binary file is a quicker and more efficient way of storing programs. Binary files are essential when saving and loading machine-code programs.

The starting address from which a binary file will be saved is stored in locations 172 and 173. These locations are loaded by the save routine, with the memory locations at which the save will begin normally set to 0 and 4, thereby pointing to the start of the Basic text area at 1024.

They can be altered by the save routine to point to any location in memory. The end address of the area of memory to be saved is stored in locations 174 and 175. Normally, when saving a Basic program, these are set to the address of the double-zero byte which

(continued on next page)

(continued from previous page)

terminates the link address. The end address can be altered to any desired location.

To change either of these addresses one cannot use the normal save routine since it automatically initialises these locations. Instead, one must write a small machine-code initialisation routine incorporating the desired operating-system subroutines. By default, a Save command will write a binary file and a Load command will read a binary file.

ASCII files are normally used to store data but they can be used to store programs. Their format is the same as that displayed on the screen or entered from the keyboard. ASCII files are created or read almost exclusively by instructions from within a Basic program. A binary file is created or read mostly by direct instructions, though the Load and Save instructions can be used within a program.

An ASCII file must first be opened with an Open statement which specifies the logical file, device number, secondary address and file name. The operating system interprets these parameters and allows the user to read or write the file to the specified device.

Data is written to an ASCII file on a particular device with a command to Print to the specified logical file number, and data is read by a Read from the logical-file command.

Tape buffer

Whereas a binary file is loaded with the contents of successive memory locations, an ASCII file is loaded with a string of variables. Storing these would require the tape to be turned on and off repeatedly, retaining a few bytes of data at a time. The Vic overcomes this by having a 192-byte tape buffer into which all data to be written to, or read from tape is loaded. Only when this buffer is full is the tape motor turned on.

Data is stored on tape in blocks of 192 bytes and since the motor is turned on and off between blocks, a two-second interval is left between blocks to allow the motor to accelerate and decelerate. The beginning of the 192-character buffer starts at address 828; the pointer to the start of the buffer is located at addresses 178 and 179; the number of characters in a buffer is stored at location 166.

These locations can be used by the programmer to control the amount of space left in a data file. If, having opened a file on cassette, the command Poke 166,191 is executed, then the contents of the tape buffer — even if empty — are loaded on to the tape. If records are kept in multiples of 191 bytes, we can very easily keep null or partially-filled records allowing future data expansion.

Whether the file being stored is binary or ASCII, the recording method used is the same and involves an encoding method peculiar to Commodore and designed to ensure maximum reliability of recording and playback. Each byte of data or program is encoded by the operating system using pulses of three distinct audio frequencies, these are: long pulses with a frequency of 1,488Hz, medium pulses at 1,953Hz and short pulses at 2,840Hz.

All these pulses are square waves with a mark-space ratio of 1:1. One cycle of a medium frequency is 256μs. in the high state and 256μs. in the low state.

The operating system takes about 9ms. to record a byte of data consisting of the eight data bits, a word-marker bit and an odd-parity bit. The data bits are either ones or zeros and are encoded by a sequence of medium and short pulses. A one is one cycle of a medium-length pulse followed by one cycle of a short-length pulse and zero is one cycle of a short-length pulse followed by one cycle of a medium-length pulse. Each bit consists of two square-wave pulse cycles, one short and one medium with a total duration of 864μs. The wave-form timing is shown in the diagram in figure 4.

The odd-parity bit is required for error checking and is encoded like the eight data bits — using a long and short pulse. Its state is determined by the contents of the eight data bits. The word marker separates each byte of data and also signals to the operating system the beginning of each byte. The word marker is encoded as one cycle of a long pulse followed by one cycle of a medium pulse, see figure 4.

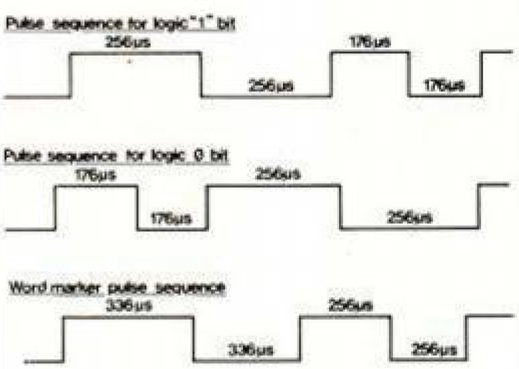


Figure 4. Operating system pulse sequences.

Since a byte of data is recorded in just 8.96ms., a 192-byte block of data in an ASCII file should be recorded in slightly more than 1.7 seconds. However, timing such a recording shows that it takes 5-7 seconds. There are two causes for this discrepancy in timing. First, to reduce the possibility of audio dropouts, the data is recorded twice. Secondly, a two-second inter-record gap is left between each record of 192 bytes.

The extensive use of error-checking techniques is one reason why the tape system on the Vic is so much better than that available on most other popular computers. There are two levels of error checking. The first divides the data into blocks of eight bytes and then computes a ninth byte, the check-sum digit. The check-sum is obtained by adding the eight data bytes together; it is the least-significant byte of the result.

On reading the tape, if one bit in the eight bytes is dropped and a zero becomes a one or vice versa, the check-sum can be used to detect this error. To do this, the same procedure to calculate the check digit is performed. The result will be different to that stored in byte 9 which is the check digit of that block computed when the tape was recorded.

The second level of error checking involves recording each block of data twice. This allows errors detected by the check digit to be corrected during the second reading of the 192-byte data block. By recording the data twice, a verification can be performed by comparing the contents of the two blocks. This will detect the few errors not detected by the check-sum.

The use of pulse sequences, rather than two frequencies as in a standard FSK (frequency-shift keying) recording, has a great advantage since it allows the operating system to compensate easily for variations in recording speed. Normally, a hardware phase-locked-loop circuit would be used to lock the system on to the correct frequencies transmitted from the tape head. The Vic, however, uses software to perform this process.

Inter-record gaps

A 10-second leader is written on the tape before recording of the data or program commences. This leader has two functions: first, it allows the tape motor to reach the correct speed, and secondly, the sequence of short pulses written on the leader is used to synchronise the read-routine timing to the timing on the tape.

The operating system can thus produce a correction factor which allows a very wide variation in tape speed without affecting reading. The system timing used to perform both reading and writing is very accurate, based as it is on the crystal-controlled system clock and timer 1 and timer 2 of VIA #2. Inter-record gaps are only used in ASCII files and their function is to allow the tape motor time to decelerate after being turned off and accelerate to the correct speed when turned on prior to a block read or write.

Each inter-record gap is approximately two seconds long and is recorded as a sequence of short pulses in the same manner as the 10-second leader. There is also a gap between blocks. When the first block of 192 bytes is recorded, it is followed by a block end-marker which consists of one single, long pulse followed by more than 50 cycles of short pulses. Then the second recording of the 192 block starts.

The first record written on the tape after the 10-second leader in both ASCII and binary files is a 192-character file-header block. The file header contains the name of the file, the starting memory location, and the end location. In an ASCII file these addresses are the beginning and end of the tape buffer; in a binary file they point to the area of memory in which the program is to be stored.

The file name can be up to 128 bytes long, the length of the file name is stored in location 183, and when read is compared with the requested file name in the Load or Open command. If the name is the same, the operating system will read the file; if different, it will search for the next 10-second inter-file gap and another header block.

The file name is stored during a read or write operation in a block memory whose starting address is stored in locations 187 and 188. When the operation is completed these are reset to point to a location in the operating system. The starting location is normally set to the beginning of the user-memory area.

The starting address is pointed to by the contents of locations 172 and 173. The end address is stored in locations 174 and 175. Normally this is the highest byte of memory occupied by the program; it can, however, be altered to point to any address, providing it is greater than the start address. ■

ACORN ATOM MACHINE-

Roy Burgin noticed several advertisements for machine-code disassemblers for the Acorn Atom. They set him thinking that a disassembler would be a useful tool, so one evening he sat down at the keyboard and set about writing one for himself.

I DECIDED to adopt the very simple approach of using a calculated Gosub — line 60 — to pick the appropriate print line. Each individual digit is printed separately after the address to avoid suppressing zeros. To help you understand the program, here are some notes:

Line 40: Sets P, Q and R to the next three bytes of code then prints the address location and the first byte at that location in hexadecimal.

Line 45: Calculates the individual hex digits of the next two bytes for later printing.

Line 50: T is set to the second hex digit of the instruction code already printed out in line 40.

Line 55: If the second digit of the instruction is 3, 7, B or F, it is not an instruction code and this section of machine code is either garbage or data. This line picks them out, see figure 1.

Line 60: Jump calculation and Gosub.

Lines 75 to 77: Subroutines to print the appropriate number of bytes after the instruction code. This is either none, one or two.

Lines 78 to 89: Subroutines to print the description following the assembly mnemonic.

Lines 100+: Initial jump locations where each line number is equal to 100 + the value of the instruction code.

All the numbers are printed in hexadecimal form for compactness, allowing each complete instruction group to be printed on one line.

It should be possible to convert this program to run on any machine which uses a 6502 processor assuming that you can understand the Atom's advanced features. For example, ? = PEEK or POKE, P. = PRINT, % = Remainder After division, a, b, c, etc., are labels for Goto or Gosub, P.&L means Print the variable L in hexadecimal form, @ is used as the field size for number printing, R. means Return. Line number 89 is interesting as it uses Boolean algebra to print a calculated forward or backward branch destination as appropriate.

Having entered and checked the program, what does one do with it? We need some machine code to disassemble; the obvious first choice is the resident operating system.

I started by exploring the various jumps and subroutines associated with input and output and discovered the section of code responsible for scrolling the screen. This routine is normally used only when a Print statement would otherwise move off the bottom of the screen.

All it does is copy each character into the line above and then blank out the bottom line. This short piece of code is very useful, especially with the Atom's ability to Link and execute machine code at any address. Add to this the other facility provided on the Atom whereby Basic variables A, X and Y are automatically transferred into the accumulator, the X register and the Y register whenever Link is used, and you can do magic. By placing the right numbers in A and Y, you can achieve the following:

- Keep up to seven lines static at the top of the screen while scrolling the lower part: see demonstration program. The Y register is normally loaded with 32 before scrolling. That is to avoid scrolling the top line into nothing. If the Basic variable Y is set equal to any number greater than this and less than 255, then the extra number of characters will be left unscrolled when used with Basic code
  LINK # FEOA
  Use in multiples of 32 to scroll whole lines.
- Keep seven or more lines static while scrolling the lower part by omitting the first section of code which scrolls the top of the screen and Linking into the section which scrolls the bottom half only — again setting the number of unscrolled characters in the Basic variable Y. This time the code does not wait for the frame refresh so it is advisable to include a Wait statement. So this time the code is
  WAIT;LINK # FE19
- Blank 32 characters from the cursor position. This uses the routine which normally blanks the bottom line of the screen but it uses for reference the cursor position which is stored in locations # DE and # DF and this means that 32 characters will be blanked from wherever the cursor happens to be. The code is:
  LINK # FE22
- Blank up to four lines from the cursor position. Again, the Basic variable Y is used to indicate the number of characters to be blanked, less one, and we Link into the code after the point where the Y register would normally be loaded. The code is:
  Y = (number of characters — 1); LINK # FE24
- Fill up to four lines from the cursor position with any chosen character. This is an extension of the last one and is very fast with four lines being filled within the frame refresh time — Basic prints only about eight characters in this time. We set the Basic variable Y to the number of characters — minus one as before — and the Basic variable A to the character code of the chosen character. Thus we have:
  Y = (number of characters — 1)
  A = (the chosen character code value)
  Link # FE26

To see just how fast this routine is, try typing:

```
Y = 127
DO A = RND%128;WAIT;LINK FE26; U.0
```

After pressing Return, this totally useless piece of code will cause the four lines following it to be completely filled with a random character and then filled, over and over again, about 30 times per second — so fast that you cannot recognise any of the characters.

A more useful method of using this routine can be achieved by directing the position of

Figure 1. Atom op-codes, showing number of bytes.

2nd Digit (columns) / **1st Digit** (rows)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	BRK 1	ORA (I,X) 2				ORA Z 2	ASL Z 2		PHP 1	ORA IM 2	ASLA 1			ORA ABS 3	ASL ABS 3	
1	BPL 2	ORA (I),Y 2				ORA Z,X 2	ASL Z,X 2		CLC 1	ORA A,Y 3				ORA A,X 3	ASL A,X 3	
2	JSR 3	AND (I,X) 2			BIT Z 2	AND Z 2	ROL Z 2		PLP 1	AND IM 2	ROLA 1		BIT ABS 3	AND ABS 3	ROL ABS 3	
3	BMI 2	AND (I),Y 2				AND Z,X 2	ROL Z,X 2		SEC 1	AND A,Y 3				AND A,X 3	ROL A,X 3	
4	RTI 1	EOR (I,X) 2				EOR Z 2	LSR Z 2		PHA 1	EOR IM 2	LSRA 1		JMP ABS 3	EOR ABS 3	LSR ABS 3	
5	BVC 2	EOR (I),Y 2				EOR Z,X 2	LSR Z,X 2		CLI 1	EOR A,Y 3				EOR A,X 3	LSR A,X 3	
6	RTS 1	ADC (I,X) 2				ADC Z 2	ROR Z 2		PLA 1	ADC IM 2	RORA 1		JMP IND 3	ADC ABS 3	ROR ABS 3	
7	BVS 2	ADC (I),Y 2				ADC Z,X 2	ROR Z,X 2		SEI 1	ADC A,Y 3				ADC A,X 3	ROR A,X 3	
8		STA (I,X) 2			STY Z 2	STA Z 2	STX Z 2		DEY 1		TXA 1		STY ABS 3	STA ABS 3	STX ABS 3	
9	BCC 2	STA (I),Y 2			STY Z,X 2	STA Z,X 2	STX Z,Y 2		TYA 1	STA A,Y 3	TXS 1			STA A,X 3	3	
A	LDY 2	LDA (I,X) 2	LDX IM 2		LDY Z 2	LDA Z 2	LDX Z 2		TAY 1	LDA IM 2	TAX 1		LDY ABS 3	LDA ABS 3	LDX ABS 3	
B	BCS 2	LDA (I),Y 2			LDY Z,X 2	LDA Z,X 2	LDX Z,Y 2		CLV 1	LDA A,Y 3	TSX 1		LDY A,X 3	LDA A,X 3	LDX A,Y 3	
C	CPY 2	CMP (I,X) 2			CPY Z 2	CMP Z 2	DEC Z 2		INY 1	CMP IM 2	DEX 1		CPY ABS 3	CMP ABS 3	DEC ABS 3	
D	BNE 2	CMP (I),Y 2				CMP Z,X 2	DEC Z,X 2		CLD 1	CMP A,Y 3				CMP A,X 3	DEC A,X 3	
E	CPX 2	SBC (I,X) 2			CPX Z 2	SBC Z 2	INC Z 2		INX 1	SBC IM 2	NOP 1		CPX ABS 3	SBC ABS 3	INC ABS 3	
F	BEQ 2	SBC (I),Y 2				SBC Z,X 2	INC Z,X 2		SED 1	SBC A,Y 3				SBC A,X 3	INC A,X 3	

MNEMONIC OPERATION
No of BYTES

CODE MAGIC

BY ROY BURGIN

Listing 1. Atom disassembler.

```
10 IN. "START POINT"S."END POINT"E
20 @ = 1; U=16
30 FOR L=S TO E
40 P=?L;O=L?1;R=L?2;P. "&L" "P/U P%U
45 V=O/U;O=O%U;X=R/U;Y=R%U
50 T=P%U
55 IF (T+1) %4=0;GOS. d;G.65
60 GOS. (100+P)
65 NEXT L
70 END
75aP."        ";R.
76bP." "&V  W"    ";L=L+1;R.
77cP." "&V W" " X Y " ";L=L+2;R.
78dP."      DATA";R.
80eP." ("&V W",X) ";R.
81fP." ("&V W") ,Y";R.
82gP." "&X Y V W",X";R.
83hP." "&X Y V W",Y";R.
84iP." "&V W",X";R.
85jP." "&V W",Y";R.
86kP." "&X Y V W;R.
87lP." "&V W;R.
88mP."@"&V W;R.
89nP." "&V W" (TO "&(L+(O<128)*(O+1)+(O>127)*(O-255))")";R.
100 GOS.a;P."BRK";R.
101 GOS.b;P."ORA";G.e
102 G.d
104 G.d
105 GOS.b;P."ORA";G.l
106 GOS.b;P."ASL";G.l
108 GOS.a;P."PHP";R.
109 GOS.b;P."ORA";G.m
110 GOS.a;P."ASL A";R.
112 G.d
113 GOS.c;P."ORA";G.k
114 GOS.c;P."ASL";G.k
116 GOS.b;P."BPL";G.n
117 GOS.b;P."ORA";G.f
118 G.d
120 G.d
121 GOS.b;P."ORA";G.i
122 GOS.b;P."ASL";G.i
124 GOS.a;P."CLC";R.
125 GOS.c;P."ORA";G.h
126 G.d
128 G.d
129 GOS.c;P."ORA";G.g
130 GOS.c;P."ASL";G.g
132 GOS.c;P."JSR";G.k
133 GOS.b;P."AND";G.e
134 G.d
136 GOS.b;P."BIT";G.l
137 GOS.b;P."AND";G.l
138 GOS.b;P."ROL";G.l
140 GOS.a;P."PLP";R.
141 GOS.b;P."AND";G.m
142 GOS.a;P."ROL A";R.
144 GOS.c;P."BIT";G.k
145 GOS.c;P."AND";G.k
146 GOS.c;P."ROL";G.k
148 GOS.b;P."BMI";G.n
149 GOS.b;P."AND";G.f
150 G.d
152 G.d
153 GOS.b;P."AND";G.i
154 GOS.b;P."ROL";G.i
156 GOS.a;P."SEC";R.
157 GOS.c;P."AND";G.h
158 G.d
160 G.d
161 GOS.c;P."AND";G.g
162 GOS.c;P."ROL";G.g
164 GOS.a;P."RTI";R.
165 GOS.b;P."EOR";G.e
166 G.d
168 G.d
169 GOS.b;P."EOR";G.l
170 GOS.b;P."LSR";G.l
172 GOS.a;P."PHA";R.
173 GOS.b;P."EOR";G.m
174 GOS.a;P."LSR A";R.
176 GOS.c;P."JMP";G.k
177 GOS.c;P."EOR";G.k
178 GOS.c;P."LSR";G.k
180 GOS.b;P."BVC";G.n
181 GOS.b;P."EOR";G.f
182 G.d
184 G.d
185 GOS.b;P."EOR";G.i
186 GOS.b;P."LSR";G.i
188 GOS.a;P."CLI";R.
189 GOS.c;P."EOR";G.h
190 G.d
192 G.d
193 GOS.c;P."EOR";G.g
194 GOS.c;P."LSR";G.g
196 GOS.a;P."RTS";R.
197 GOS.b;P."ADC";G.e
198 G.d
200 G.d
201 GOS.b;P."ADC";G.l
202 GOS.b;P."ROR";G.l
204 GOS.a;P."PLA";R.
205 GOS.b;P."ADC";G.m
206 GOS.a;P."ROR A";R.
208 GOS.c;P."JMP ("&X Y V W")";R.
209 GOS.c;P."ADC";G.k
210 GOS.c;P."ROR";G.k
212 GOS.b;P."BVS";G.n
213 GOS.b;P."ADC";G.f
214 G.d
216 G.d
217 GOS.b;P."ADC";G.i
218 GOS.b;P."ROR";G.i
220 GOS.a;P."SEI";R.
221 GOS.c;P."ADC";G.h
222 G.d
224 G.d
225 GOS.c;P."ADC";G.g
226 GOS.c;P."ROR";G.g
228 G.d
229 GOS.b;P."STA";G.e
230 G.d
232 GOS.b;P."STY";G.l
233 GOS.b;P."STA";G.l
234 GOS.b;P."STX";G.l
236 GOS.a;P."DEY";R.
237 G.d
238 GOS.a;P."TXA";R.
240 GOS.c;P."STY";G.k
241 GOS.c;P."STA";G.k
242 GOS.c;P."STX";G.k
244 GOS.b;P."BCC";G.n
245 GOS.b;P."STA";G.f
246 G.d
248 GOS.b;P."STY";G.i
249 GOS.b;P."STA";G.i
250 GOS.b;P."STX";G.j
252 GOS.a;P."TYA";R.
253 GOS.c;P."STA";G.h
254 GOS.a;P."TXS";R.
256 G.d
257 GOS.c;P."STA";G.g
258 G.d
260 GOS.b;P."LDY";G.m
261 GOS.b;P."LDA";G.e
262 GOS.b;P."LDX";G.m
264 GOS.b;P."LDY";G.l
265 GOS.b;P."LDA";G.l
266 GOS.b;P."LDX";G.l
268 GOS.a;P."TAY";R.
269 GOS.b;P."LDA";G.m
270 GOS.a;P."TAX";R.
272 GOS.c;P."LDY";G.k
273 GOS.c;P."LDA";G.k
274 GOS.c;P."LDX";G.k
276 GOS.b;P."BCS";G.n
277 GOS.b;P."LDA";G.f
278 G.d
280 GOS.b;P."LDY";G.i
281 GOS.b;P."LDA";G.i
282 GOS.b;P."LDX";G.j
284 GOS.a;P."CLV";R.
285 GOS.c;P."LDA";G.h
286 GOS.a;P."TSX";R.
288 GOS.c;P."LDY";G.g
289 GOS.c;P."LDA";G.g
290 GOS.c;P."LDX";G.h
292 GOS.b;P."CPY";G.m
293 GOS.b;P."CMP";G.e
294 G.d
296 GOS.b;P."CPY";G.l
297 GOS.b;P."CMP";G.l
298 GOS.b;P."DEC";G.l
300 GOS.a;P."INY";R.
301 GOS.b;P."CMP";G.m
302 GOS.a;P."DEX";R.
304 GOS.c;P."CPY";G.k
305 GOS.c;P."CMP";G.k
306 GOS.c;P."DEC";G.k
308 GOS.b;P."BNE";G.n
309 GOS.b;P."CMP";G.f
310 G.d
312 G.d
313 GOS.b;P."CMP";G.i
314 GOS.b;P."DEC";G.i
316 GOS.a;P."CLD";R.
317 GOS.c;P."CMP";G.h
318 G.d
320 G.d
321 GOS.c;P."CMP";G.g
322 GOS.c;P."DEC";G.g
324 GOS.b;P."CPX";G.m
325 GOS.b;P."SBC";G.e
326 G.d
328 GOS.b;P."CPX";G.l
329 GOS.b;P."SBC";G.l
330 GOS.b;P."INC";G.l
332 GOS.a;P."INX";R.
333 GOS.b;P."SBC";G.m
334 GOS.a;P."NOP";R.
336 GOS.c;P."CPX";G.k
337 GOS.c;P."SBC";G.k
338 GOS.c;P."INC";G.k
340 GOS.b;P."BEQ";G.n
341 GOS.b;P."SBC";G.f
342 G.d
344 G.d
345 GOS.b;P."SBC";G.i
346 GOS.b;P."INC";G.i
348 GOS.a;P."SED";R.
349 GOS.c;P."SBC";G.h
350 G.d
352 G.d
353 GOS.c;P."SBC";G.g
354 GOS.c;P."INC";G.g
```

the cursor by altering the values stored in locations # DE and # DF, or if you prefer, 222 and 223. Try this:

```
Y=127
A=127
FOR I=#8000 TO #8180 STEP 128;!222=I;I
  LINK#FE26;NEXT I
```

This routine will turn the screen white in a fraction of a second and appears instantaneous.

Have fun experimenting with these routines but be warned, using pieces of code like this is like reading something out of context and odd things may happen. For instance, if you cause the screen to scroll while the cursor is not at the bottom of the screen, anything already on the bottom line will not be erased but will be copied into the next line.

If you have not made the cursor invisible by typing

?# E1=0,

you may leave the odd white square at the previous cursor position.

The demonstration program given in listing 2 shows how the scrolling routine works. It is only a demonstration but it illustrates the principle satisfactorily. ■

Listing 2. Demonstration program.

```
100 P.$12"          DEMONSTRATION PROGRAM"
120 P."        PARTLY SCROLLING SCREEN "
160 REM REMOVE THE CURSOR
180 ?#E1=0
220 REM SET NUMBER OF UNSCROLLED LINES IN Y
240 REM Y=(NUMBER OF UNSCROLLED LINES+1)*32
250 REM Y=(6+1)*32=224     FOR SIX LINES
260 Y=224
300 REM PRINT SOMETHING ON ONE LINE
320sFOR I=1 TO 3
340 P.RND%(I*30)
360 NEXT I
400 REM DELAY
420 FOR I=1 TO 10;WAIT;NEXT I
460 REM SCROLL IT
480 LINK#FE0A
520 REM MOVE THE CURSOR BACK TO THE START OF THE SAME LINE
540 P.$11
580 REM GO BACK AND PRINT SOME MORE
600 GOTOs
```



ZX81 1K PROGRAMS

FROM VIDEO SOFTWARE LTD.

These programs are written to the same high standard as our highly acclaimed 16K software. All the programs are entirely original (note the absense of Mastermind, Lunar Lander, etc.). They are so good that they are worth buying even if you have the 16K RAM.

ZX81 1K PARTY TRICKS NO. 1 includes:

1. SHOOT. Pick your spot, take the penalty and watch the goalie attempt to save — sometimes he succeeds.
2. SKETCH. Use cursor keys to draw large pictures — almost full screen and save them on tape.
3. NAME THE DAY. Calculates the day of the week corresponding to any valid date in the twentieth century.
4. DRIVE THE TRAIN. Ever wanted to drive a train. This little program will amuse the youngest members of the family.
5. ONGER-WONGER. Watch the computer draw its own pictures and set it up to draw yours.
6. WEATHER. An endless variety of regional weather forecasts — all of them inaccurate but then so are the real ones.
7. UFO. The inevitable space game. Shoot down the circling UFO using your photon torpedo under keyboard control.
8. WHO SHOT J.R. Four suspects, four places, four times of day, four weapons. From the clues given deduce who shot J.R.
9. FIELD GUN. You are the officer in charge of a field gun. Can you hit the target.
10. MYSTERY PROGRAM. Still to be decided but it will be as good as the others.

Note: These programs are *NOT suitable for ZX80.*

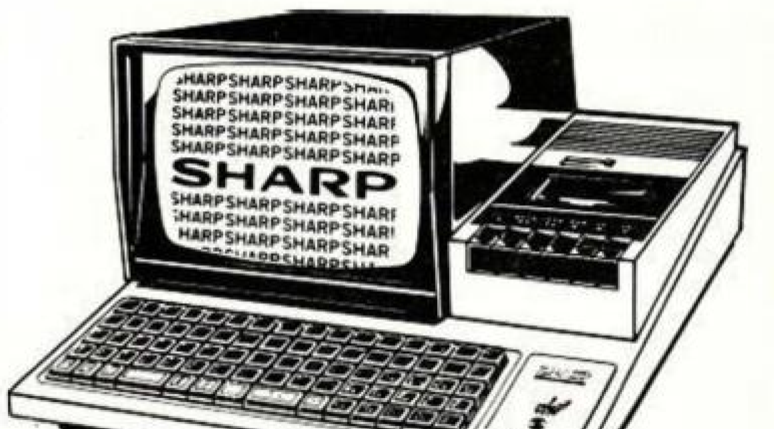
Manual only with fully annotated listing.......................£1.95
Manual and cassette.......................£4.95
De luxe boxed set — ideal gift.......................£7.95

VIDEO SOFTWARE LTD.
Stone Lane, Kinver, Stourbridge, West Midlands DY7 6EQ

Allow 28 days for delivery

SHARP MZ80K

£375 inc. VAT



COMPUTER 100 LIMITED
7 Southcote Parade
Southcode Farm Road
Reading
Berkshire
Tel: Reading 584545

48K

Mail order:
If you are unable to visit our showroom, then please add £7.50 to cover insurance and transit.

Please supply complete 48K Sharp MZ-801

NAME..

ADDRESS..

....

....

YC1/1

PROJECT

COGENT STEPS TOWAR[

In the previous part of his series, John Dawson began to outline an interpreter for the 6502 microprocessor. He continues with an analysis of the most suitable language level and some fundamental subroutines.

LAST MONTH, I discussed the difference between compiled and interpreted high-level computer languages and gave some building-block subroutines as the start of a control-orientated interpreter for the Tangerine Microtan or Micron computer. Programs need names and Cogent seems appropriate for this project.

Last year I spent some time working on a program called Duncan which was an interpreted language for the Nascom 1 micro-computer. Figure 1 is a short but typical Duncan program. Program instructions in Duncan consisted of a single letter to identify the main instruction heading followed, where necessary, by suffixed letters or numbers.

These identify either a port or label, or provide a conditional code for a Jump or Goto instruction and the label to which the program should transfer execution. When you are familiar with Duncan it is easy to use and you can build reasonably complex control procedures.

Duncan used reversed-Polish notation for its number-handling procedures based on an earlier language M5, and could cope with positive integers in the range zero to 65,535. Eight-bit — one-byte — values could be set to output devices or obtained as an input value. Duncan had no facility for handling interrupts but could be used to poll a number of devices, either continuously, or at a defined stage in the execution of a program.

A number of ideas occurred to me in the course of thinking about an improved interpreter for the Microtan: first, it seemed desirable that the source program should be written in English with as few restrictions on the author as possible. A person writing the program should be able to describe the tasks the computer would perform using plain language and full-length words.

The compacted coding used in many high-level languages is less relevant now that RAM is less expensive and this code will become increasingly obsolete as more people become familiar with typewriter keyboards. I thought that the program should be constructed along the lines of simple but recognisable English grammar.

For example, the source program should consist of a series of sentences with a constant terminating feature — a full stop at the end of the sentence followed by one or more spaces. It seemed that spaces between words would become significant since the transition from an alpha-numeric character to a space marks the end of a word. Consequently, the names of variables and input/output devices should be linked by a hyphen when they consist of more than one word.

I thought it important that the program should allow redundant material in the source program. In other words, the author should not have to formally declare that a word or a group of words is a remark to be ignored by the computer. The computer should be able to look through a source sentence and extract from it the information it needs to carry out the instructions contained in the sentence.

Third, eight-bit accuracy is entirely adequate for the vast majority of domestic applications and school or amateur experimental purposes. However, the interpreter should be able to handle large integer numbers, both positive and negative, for many tasks including signal averaging and other scientific applications.

Fourth, reverse-Polish notation is easy to work with, both in programming terms and as a user. Hewlett Packard has built a successful

```
(TEST ROUTINE — SR CALL AND JUMP
  BACK)
. (FIRST NUMBER) ?=A PL P P P
. (SECOND NUMBER) ?=B PL P.
. £7
. (FIRST PLUS SECOND=) C=? PL
JS L7 A,B+ =C :
(END)

FIRST NUMBER 479 SECOND NUMBER 513
FIRST PLUS SECOND=00992
```

Figure 1. A typical Duncan program.

range of calculators around RPN for good reasons, and I thought that I would continue to use RPN for Cogent.

There is a classic dilemma inherent in interpreted languages — if it is easy for humans to read and understand, it will be slow for the machine to interpret and execute. Duncan ran very quickly because the Nascom had only to examine a single character to jump to the next instruction group.

Identifying an instruction from a variable-length word, which requires the computer to match a list of instructions and list of input and output devices, might slow the program considerably. It could even be slowed to the point where, while still acceptable for many domestic control jobs, it would be useless for acquiring data from many experiments in a school physics laboratory.

The task of writing an interpreter which would compress a source program to single-byte instructions for execution by the machine, re-expanding the compacted code to the full source listing for editing purposes is not for a part-time amateur systems analyst and programmer — however dedicated or obsessed.

What is much more important in the design of a language is the framework it provides for describing the problem faced by the analyst. In other words, a good programming language should be capable of helping a user to describe what he wishes the computer to achieve — it should lead him through the design of a program along a logical route.

Some of the newer languages such as Pascal and APL have comparatively simple, coherent designs which influence the programmer's perception of the problem. In this way, the expressions Do-Until or Repeat-While describe not just a feature of a programming language but a useful and powerful approach to investigating and writing a solution to a problem.

Let us now examine an important part of any control-orientated program — a real-time clock. I used the two-pass assembler from Microtanic Software to write the clock subroutines and it is an enormous advance on the translator/disassembler in the Tangerine XBug.

The good features of the Microtanic product are that: the assembler is very fast; the second, and longer, pass through more than 200 lines of source code took just 4.5 seconds. There is a high degree of control over the positioning of source and object code and it is possible to assemble directly form a source-code tape with the result that the whole of the Microtan memory can be filled with object code.

The pseudo-ops BYT and WOR allow the use of labels as operands which considerably enhances the assembler because it makes the object code truly relocatable. There is a tantalising mention in the documentation of further pseudo-op codes to allow the use of macro instructions in the future.

On the debit side, my comments are mostly to do with the documentation. It is adequate, but more examples would have been helpful and the layout could have been improved to make some difficult concepts more easily understood. The documentation is printed on a dot-matrix printer with no descenders. Software of this quality justifies a higher standard of presentation.

One section describing the use of labels as operands to the BYT and WOR pseudo-ops

S PROCESS CONTROL

still gives me problems even though I used the feature successfully in programming the real-time clock.

The assembler controls a printer, somewhat clumsily, either by internal software in the EPROM or by your own external program. However, the printer can only be used during the second-pass assembly or to list the source-program labels, and I could not find a way of listing the source program alone.

These criticisms are minor — the Tansoft two-pass assembler is an effective and powerful piece of work, backed up by more than 5,000 words of explanation and support. The three sections of the documentation — "Installations", "What is a two-pass assembler?" and "All the technical stuff" — will start you off on the right track if you are prepared to concentrate. The results of using the assembler are shown in figure 2.

The timing values £$40 in line 0053 and £$8B in line 0055 may need adjustment to the crystal frequency in your own machine. At present, the program will measure only elapsed time from the moment when the program starts to run. I hope to list another short routine to set the real time in next month's article.

The clock records the week of the year, day of the week, as well as hours, minutes and seconds. When the program is executed at 0400 hex, there should be a one-second delay and then the top line of the screen is partially cleared and the following figures should be displayed:

00-00 00:00:01

The figures read from the left according to the following key:

Week(0-51) — Day(0—6)
Hours(24):Minutes:Seconds

Every 50ms. the VIA counter reaches zero and generates an interrupt. The VIA timer is then automatically reloaded with the initial values and the timing cycle recommences. The interrupt diverts the 6502 CPU from whatever program it is executing to the clock interrupt CLKINT routine starting at line 107. In the course of updating the clock counters, the routine displays the current time using the CLODIS and B2D subroutines. The time is updated every second which explains the initial delay when the program is started.

The instruction LSR @ is equivalent to LSR A and is peculiar to the Tansoft two-pass assembler. With this exception, all the mnemonics are standard 6502 assembly-language instructions. Check the machine code corresponding to the assembly instruction if you have any difficulty.

The final program section at line 183 initialises the clock and starts it running. The program then loops and will look for an input from the keyboard to display on the VDU. ■

Figure 2. Clock subroutines generated by the Microtanic Software two-pass assembler.

```
0001 ;                            0400
0002 ;                            0400
0003 ;     COGENT                 0400
0004 ;                            0400
0005 ;    by JOHN DAWSON          0400
0006 ;    October 1981            0400
0007 ;                            0400
0008 ;                            0400
0009 ;                            0400
0010          JMP START           0400 4C 01 05
0011 ;                            0403
0012 ;                            0403
0013 ;                            0403
0014 ;                            0403
0015 ; VECTORS                    0403
0016 ;                            0403
0017 CKVEC   WOR CLKINT           0403 8F 04
0018 ;                            0405
0019 CFACE   EPZ $40              0405
0020 WK      EPZ $F8              0405
0021 DY      EPZ $F9              0405
0022 HH      EPZ $FA              0405
0023 MM      EPZ $FB              0405
0024 SS      EPZ $FC              0405
0025 MS      EPZ $FD              0405
0026 CFIND   EPZ $FE              0405
0027 ;                            0405
0028 ;                            0405
0029 ; CLEAR CLOCK COUNTERS       0405
0030 ;                            0405
0031 CLIN    LDA £$0              0405 A9 00
0032         TAX                  0407 AA
0033 CB1     STA WK,X             0408 95 F8
0034         INX                  040A E8
0035         CPX £$07             040B E0 07
0036         BCC CB1              040D 90 F9
0037         LDA £20              040F A9 14
0038         STA MS               0411 85 FD
0039 ;                            0413
0040 ; SET UP VIA                 0413
0041 ;                            0413
0042         SEI                  0413 78
0043         LDA £$4C             0414 A9 4C
0044         STA $10              0416 85 10
0045         LDA CKVEC            0418 AD 03 04
0046         STA $11              041B 85 11
0047         LDA CKVEC+1          041D AD 04 04
0048         STA $12              0420 85 12
0049         LDA £$C0             0422 A9 C0
0050         STA $BFCE            0424 8D CE BF
0051         LDA £$40             0427 A9 40
0052         STA $BFCB            0429 8D CB BF
0053         LDA £$40             042C A9 40
0054         STA $BFC6            042E 8D C6 BF
0055         LDA £$8B             0431 A9 8B
0056         STA $BFC5            0433 8D C5 BF
0057         LDA £$0D             0436 A9 0D
0058         STA CFACE            0438 85 40
0059         LDA £$02             043A A9 02
0060         STA CFACE+1          043C 85 41
0061         CLI                  043E 58
0062         RTS                  043F 60
0063 ;                            0440
0064 ;                            0440
0065 ; CLOCK DISPLAY SR           0440
0066 ;                            0440
0067 CLODIS LDY £$08              0440 A0 08
0068         LDA £$02             0442 A9 02
0069         STA $4F              0444 85 4F
0070         LDA £$00             0446 A9 00
0071         STA $4E              0448 85 4E
0072 CD1     LDA £$20             044A A9 20
0073         STA ($4E),Y          044C 91 4E
0074         INY                  044E C8
0075         CPY £$20             044F C0 20
0076         BNE CD1              0451 D0 F7
0077         LDA £$00             0453 A9 00
0078         STA CFIND            0455 85 FE
0079         LDA WK               0457 A5 F8
0080         JSR B2D              0459 20 E7 04
0081         LDY CFIND            045C A4 FE
0082         LDA £$2D             045E A9 2D
0083         STA (CFACE),Y        0460 91 40
0084         INC CFIND            0462 E6 FE
0085         LDA DY               0464 A5 F9
0086         JSR B2D              0466 20 E7 04
0087         INC CFIND            0469 E6 FE
0088         INC CFIND            046B E6 FE
0089         INC CFIND            046D E6 FE
0090         LDA HH               046F A5 FA
0091         JSR B2D              0471 20 E7 04
0092         LDY CFIND            0474 A4 FE
0093         LDA £$3A             0476 A9 3A
0094         STA (CFACE),Y        0478 91 40
0095         INC CFIND            047A E6 FE
0096         LDA MM               047C A5 FB
0097         JSR B2D              047E 20 E7 04
0098         LDA £$3A             0481 A9 3A
0099         LDY CFIND            0483 A4 FE
0100         STA (CFACE),Y        0485 91 40
0101         INC CFIND            0487 E6 FE
0102         LDA SS               0489 A5 FC
0103         JSR B2D              048B 20 E7 04
0104         RTS                  048E 60
0105 ;                            048F
0106 ;                            048F
0107 ;  CLOCK INTERRUPT            048F
0108 ;                            048F
0109 CLKINT PHA                   048F 48
0110         TXA                  0490 8A
0111         PHA                  0491 48
0112         TYA                  0492 98
0113         PHA                  0493 48
0114         DEC MS               0494 C6 FD
0115         BPL EX2              0496 10 45
0116         SED                  0498 F8
0117         CLC                  0499 18
0118         LDA SS               049A A5 FC
0119         ADC £01              049C 69 01
0120         STA SS               049E 85 FC
0121         CMP £$60             04A0 C9 60
0122         BCC RES20            04A2 90 32
0123         LDA £$00             04A4 A9 00
0124         STA SS               04A6 85 FC
0125         ADC MM               04A8 65 FB
0126         STA MM               04AA 85 FB
0127         CMP £$60             04AC C9 60
0128         BCC RES20            04AE 90 26
0129         LDA £$00             04B0 A9 00
0130         STA MM               04B2 85 FB
0131         ADC HH               04B4 65 FA
0132         STA HH               04B6 85 FA
0133         CMP £$24             04B8 C9 24
0134         BCC RES20            04BA 90 1A
0135         LDA £$00             04BC A9 00
0136         STA HH               04BE 85 FA
0137         ADC DY               04C0 65 F9
0138         STA DY               04C2 85 F9
0139         CMP £$07             04C4 C9 07
0140         BCC RES20            04C6 90 0E
0141         LDA £$00             04C8 A9 00
0142         STA DY               04CA 85 F9
0143         ADC WK               04CC 65 F8
0144         CMP £$52             04CE C9 52
0145         BCC RES20            04D0 90 04
0146         LDA £$00             04D2 A9 00
0147         STA WK               04D4 85 F8
0148 RES20   LDA £20              04D6 A9 14
0149         STA MS               04D8 85 FD
0150         JSR CLODIS           04DA 20 40 04
0151 EX2     LDA $BFC4            04DD AD C4 BF
0152         CLD                  04E0 D8
0153         PLA                  04E1 68
0154         TAY                  04E2 A8
0155         PLA                  04E3 68
0156         TAX                  04E4 AA
0157         PLA                  04E5 68
0158         RTI                  04E6 40
0159 ;                            04E7
0160 ;                            04E7
0161 ; CONVERT TIME FOR DISPLAY04E7
0162 ;                            04E7
0163 B2D     PHA                  04E7 48
0164         LDX £$01             04E8 A2 01
0165         LDY CFIND            04EA A4 FE
0166         LSR @                04EC 4A
0167         LSR @                04ED 4A
0168         LSR @                04EE 4A
0169         LSR @                04EF 4A
0170 PNT1    CLC                  04F0 18
0171         ADC £$30             04F1 69 30
0172         STA (CFACE),Y        04F3 91 40
0173         INY                  04F5 C8
0174         DEX                  04F6 CA
0175         BEQ MORE             04F7 F0 03
0176         STY CFIND            04F9 84 FE
0177         RTS                  04FB 60
0178 MORE    PLA                  04FC 68
0179         AND £$0F             04FD 29 0F
0180         BPL PNT1             04FF 10 EF
0181 ;                            0501
0182 ;                            0501
0183 START   JSR CLIN             0501 20 05 04
0184 MAIN1   JSR $FDFA            0504 20 FA FD
0185         LDA $01              0507 A5 01
0186         JSR $FE75            0509 20 75 FE
0187         CLC                  050C 18
0188         BCC MAIN1            050D 90 F5
```

RESPONSE FRAME

Do you have a problem? Your manual is incomprehensible or you just cannot get the hang of that programming trick you tried — whatever it is, Tim Hartnell will do his best to answer your queries. Please include only one question per letter and mark them "Response Frame".

CODE BOOKS

■I should very much like to know if any books exist dealing with the vexed question of ZX-80 machine code — preferably for beginners. I have seen the odd machine-code routine, but no explanation about how to enter it, or reserve space in the RAM, and little to explain what the Pushing and Banking actually do.

Michael Pout,
Bexley, Kent.

BOOKS WHICH could help you include: *Machine language programming made simple*, £8.95 and *Understanding your ZX-81 ROM*, by Dr Ian Logan, £8.95. Both are available from The Essential Software Company, 47 Brunswick Centre, London, WC1. *Mastering machine code on your ZX-81 or ZX-80*, Tony Baker, £5.95 is available from Interface, 44-46 Earls Court Road, London, W8 6EJ.

CHESS CHOICE

■Interested in microcomputers and as the owner of a chess computer, I particularly enjoyed the article by John White on chess computers in the August/September issue. I wish to exchange my chess computer for a more powerful model and thought it might be a good idea to buy a microcomputer with such a chess-playing capability. John White stated that the Morphy 2.5 program was written on only 8K ROM and 1K RAM. The Sinclair ZX-81 also is advertised as having 8K ROM and 1K RAM. Would I be mistaken in thinking that such a powerful program as the Morphy 2.5 could be written for use on the Sinclair ZX-81?

J S Strange,
Cockett, Swansea.

AS A GENERAL rule, I would suggest that a chess program for a microcomputer is unlikely to play as well as a chess machine. The Morphy program you refer to is written entirely for chess, that is, the memory is totally dedicated to chess, whereas the same memory on, say, a ZX-81 is required to do many, many things. When a chess program is used with a computer, it uses only part of the computer's capabilities. John White suggests in his article that the Sargon 2.5 is the best chess-playing machine available, and — if you are serious about chess — is a better buy for that purpose than buying a computer and then buying a chess program for it. However, when you have a computer, you can, of course, do far more with it than just play chess. One shop which carries a number of chess computers, where you can compare their levels of play, is the Silica Shop, 1-4 The Mews, Hatherley Road, Sidcup, Kent; telephone 01-301 1111.

ZX-81 BUG TEST

WE HAVE RECEIVED a number of questions about a bug in the ZX-81 ROM. The bug appears when dealing with very small, or very large numbers, and is especially evident with some log functions, and raising numbers less than one — such as squaring 0.125 — to a power. The bug in the ROM has been located. We believe three extraneous lines in the original ROM have simply been deleted, a discovery made by Frank O'Hara and comfirmed by Ian Logan. Sinclair Research is now swapping old ROMs for new. If you ring Nigel Brown on 0276-62155, he will tell you how to go about obtaining a modified ROM. If you decide to buy a ZX-81 from W H Smith, test it first to see if it has a ROM-with-bug by entering the following line:

```
PRINT .125**2
```

Any answer other than the correct one — .015625 — shows the computer has an old ROM in it. Sinclair Research tell us that only new ROMs are now on the market, but it has admitted that a few old ROMs may still be in circulation.

POOLS SYSTEM

■I was delighted to read your first issue, particularly the ZX-81 review. Unlike others, you do not hesitate to find fault, and I trust you for this attitude. I am interested in football pools, and to date have laboriously conducted research by hand, which covers the data of 15 seasons with 42 matches on each of the season's 55 playing days — no mean effort. I would like your opinion as to the best computer to aim at. Obviously I cannot run to an IBM or similar. Would you care to make any suggestions, please?

Robert Mason,
Pickering, North Yorkshire.

YOU DO NOT actually say what you wish to do with the computer in relation to the pools, but we imagine you wish to be able to compare, analyse and predict. The volume of data is rather large, but could be handled in one-year chunks by a ZX-81 with 16K of memory, or any small microcomputer of similar capacity. However, to handle all 15 years at once, to be able to call anything from those 15 years at will, would require much more memory. A disc-operating system would possibly be an idea, if you really must have all 15 years on tap at any one time. A £7 football-pools program is available for the ZX-80 from: Peter Vasey, 18 Ferndale Grove, East Boldon, Tyne and Wear.

VIC DOWN UNDER

■I would like to pose a question regarding the Vic-20. I read in your magazine the Vic-20 has to be tuned to suit the differing sound channels and TV standards. I will be buying a Vic-20 soon, and hope to build it into a larger system with all the peripherals. The trouble is that in a few years I may go to Australia. Would I be able to adjust the Vic-20 to suit the TV or would I have to sell the system?

Paul Ormonde-Jones,
St Albans, Hertfordshire.

THE AUSTRALIAN television system is fully compatible with the British system, so you will have no problems. You can find out more about the Vic from distributors, which include the Byteshop, 01-387 0505, and The Vic Centre, 01-992 9904.

BOARD CONTACT

■I am interested in chess programming, but do not know where to start. I am a programmer by profession and my favourite languages are Basic, Cobol and Fortran. Please let me know of anyone who has written chess programs in this part of the world. Perhaps we could meet and help each other.

John Kay,
Lagos, Nigeria.

WE DO NOT know of anyone in your part of the globe who has worked on chess programs, but suggest that one way to contact people who have done so in the U.K. would be to write to a few of the smaller firms advertising chess programs. These are most likely to have been written by the people running the company, and they may be able to help you. Also, you could buy some of these programs to analyse them. Doing this may give you some ideas for writing your own.

COMPROMISE

■During the past month I have had limited access to a Commodore Pet. During this time I have acquired some programming skills, and I would now like to buy a micro for use at home. The Pet is out of my price range. Perhaps you would advise me about which computer would incorporate the features listed and would give me the best value for money:

- **Reliability with software available.**
- **Cassette loading using existing cassette player.**
- **Expandable memory.**
- **Good format size with good graphics for drawing purposes.**
- **Should cost not more than £200.**

A Birch,
Tuffley, Gloucester.

UNFORTUNATELY, your demands are incompatible. The Vic-20 would enable you to use, almost without modification, the programming skills you have picked up on the Pet and also has expandable memory. There is software available for it and the machine costs around £200. However, you need the special Vic cassette player — £40 to £50 — and the graphics as supplied are coarse. If you are prepared to spend up to £300 or £350, a number of suitable machines are available, including the BBC Micro, Tangerine, Video Genie and the like. If you have limited money, your purchase will have to be a compromise. ■

SANITY SAVER

■Can you please save my sanity? I am 14 and sold my ZX-80 hoping to save up for a Vic-20, when a friend suggested I buy the whole ZX-81 kit — memory, printer, computer — which would be the same price. Another friend proposed an Atom or UK101, and my father advocated a Superboard. Please could you help and guide me?

D J Marsh,
Cadishead, Manchester.

RODNEY ZAKS, of Sybex tells *Your Computer* that the trend in America nowadays is for people to work out what they want to do with their computer, and then buy one which fits that specification. We can only suggest you do the same. The questions you will need to have clear in your mind when deciding what to buy could include:

- How much money do I have to spend?
- What graphics quality do I want?
- What do I want to do with it? — for example, play Space Invaders; program in machine code, in Basic; control external devices like lights.
- How much memory will I need for this?
- Will I be likely to want to add a printer or other peripherals?
- Does it need an external cassette machine or has it one built in, or will it require floppies?
- If the computer generates a colour picture, do I have access to a colour TV to make use of the colour facility?

Ask your self these questions, putting the answers in writing, then try and find a machine to match your answers. ■

WILTSHIRE

SPECIAL OPENING OFFER

* LOWEST PRICE
* SHARP MZ-80K

20K Machine only **£365.00**
16K RAM upgrade (including fitting) **£42.50**
Also in stock — VIDEO GENE at **£299.00**
VIDEO GENE II at **£235.00**
Sound Kit **£10.00** Colour Board Kit **£35.00**
D.A.I. Colour Computer — 48K + Stereo Sound only **£595.00**

Centronix 737 Printer **£359.95**
Software — Books — Service

Orders accepted by telephone using Access or Barclaycard. Postage £9.00.
Credit terms are available.

Please add 15% VAT to all prices.

Orders now being taken for Vic 20

VISIT the Friendly Computer Shop

Everyman Computers

14 EDWARD STREET, WESTBURY, WILTS.
Tel: (0373) 864644 Home Tel: 823764

Open all day Saturday

ZX81 owners

Protos



Keyboard is here!

At last! A *real*, full size keyboard in a top quality case for your ZX-81.

Simply unscrew your ZX-81 printed circuit board from its black Sinclair case and plug into Protos.

FULLY BUILT £64.95 inc. VAT

- More accurate, faster typing with bigger and *real* keys
- 40 colour coded key-tops for easy reading
- Robust, 'big' computer construction
- PCB prepared for more add-ons to come
- New edge connector provided for Sinclair and other manufacturers' peripherals
- Key legends can be changed for future new ROM functions
- Sinclair PCB fully enclosed — and room for much more

If you feel you've outgrown your ZX-81 don't sell it for peanuts and move to another system. Add it to Protos and make your ZX-81 grow.

For details, large SAE, please. For orders add £2.50 post and packing. Cheques to 'Frome Computing'.

Protos Computer Systems

Frome Computing,
20 Ashtree Road, Frome,
Somerset,
BA11 2SF

MORE POWER TO YOUR POCKET!



Hand held alphanumeric programmable-BASIC language-holds up to ten different programs simultaneously-subroutines nested up to ten levels-program looping up to eight levels-simplified program editing and debugging-variable programming capacity: between 1680 steps with 26 memories and 80 steps with 226 memories-55 single key routines including log, trig and hyperbolic-built-in routines include standard deviation (both types), regression analysis and correlation coefficient-all programs and memory data retained even when switched off. *Comprehensive library with over 70 program examples. Optional FA2 adaptor for program storage on cassette and FP10 printer.*

THE FX702P. A pocket computer that communicates in BASIC language.

R.R.P. £134.95

AVAILABLE AT SPECIALIST CASIO CALCULATOR OUTLETS.

CASIO WHAT WILL THEY THINK OF NEXT?

CASIO ELECTRONICS CO. LTD., SHAIBURN HOUSE, 28 SCRUTTON STREET, LONDON EC2A 4TY.

FINGERTIPS

Fingertips is our regular calculator column covering calculator news, programming hints and examples of unusual applications. The column is written and compiled by calculator enthusiast David Pringle who is glad to hear of any of your ideas. *Your Computer* pays £6 for each of your contributions published.

FOR CASIO TO release its third major line of calculators in a year — as it has done with the Casio 702-P, takes some organisation, in both senses of the word. The 702 provides the first opposition to the Sharp PC-1211 and Tandy programmables in that it uses Basic and has similar capabilities. Table 1 gives a check list of the more important characteristics.

Robin Terry of Barking provides us with our first Basic program of the column: Kalah — or Owani — is a game which is played widely in Africa, he writes. The playing area consists of two rows of six pits in the ground, each filled with five stones. There are two pits to the side, such that the board looks like this:

```
          Player 1
           oooooo
     O     123456     O
           oooooo
          Player 2
```

The large pit on the player's right is called his Kalah. To play, a player chooses a small pit on his side, and removes all the stones, holding them in his hand. Then, one by one, moving anti-clockwise he drops the stones in successive pits — including his own, but not his opponent's Kalah. If he holds enough stones, he will drop them into his opponent's smaller pits also. Then his opponent follows suit.

If, however, the last stone a player drops falls into his own Kalah, he then earns another turn. Also, if the last stone falls into an small, empty pit on his own side, he may empty the small pit directly opposite on his opponent's side and, together with the winning stone, drop them into his own Kalah. The game continues in this way until a player cannot play; then the player with most stones in his Kalah wins.

In reply to my request for information on methods of matrix inversion, Boris Allan of Stockport writes: It has been found by a correspondent to Fingertips — Billy Wadsworth, *Your Computer* August/September — that the Cholesky method of matrix inversion is not always satisfactory — sometimes there is an attempt to take the square root of a negative number. The Cholesky decomposition method is used to invert symmetric, positive definite matrices — that is, matrices which are symmetric and have a positive, non-zero determinant.

To attempt to take the square root of a negative number during the decomposition suggests that the determinant is negative or zero. In this situation one can exercise various options, of which the three most popular are:

- Give up
- Use a ridge method
- Use a pseudo-inverse method

Let us reject the first possibility and try the third.

Pseudo-inverse methods are rather more sophisticated in a mathematical sense. A compact routine to produce a pseudo-inverse — using the Jordan method — is given in Magi Test 1 Floating "The true value of benchmarks", *Practical Computing,* June 1981. As Billy Wadsworth probably uses the Cholesky routine to economise on storage — and the Jordan routine requires a full matrix — I will now give an algorithm for a pseudo-inverse routine, using the Cholesky method — but what is a pseudo-inverse?

This matrix is singular, because column 1 and column 3 are linearly dependent — the numbers in column 3 are twice those in column 1:

A.				
	4	5	8	2
	5	6.5	10	1.5
	8	10	16	4
	2	1.5	4	6

A pseudo-inverse routine would step through this matrix, performing various computations, until it reached column 3. If the routine were a normal Cholesky procedure, it would take the square root of zero and then try to divide by the zero value, or — with rounding errors — it might take the square root of a negative number.

The pseudo-inverse routine, however, sets the appropriate row and column to zero, so that the original matrix becomes

B.				
	4	5	0	2
	5	6.5	0	1.5
	0	0	0	0
	2	1.5	0	6

and what is really inverted is the submatrix

C.			
	4	5	2
	5	6.5	1.5
	2	1.5	6

and the inverse of this matrix is

D.			
	36.5	−27	−5.5
	−27	20	4
	−5.5	4	1

This inverse is then substituted back into the modified matrix, B, to give the pseudo-inverse

E.				
	36.5	−27	0	−5.5
	−27	20	0	4
	0	0	0	0
	−5.5	4	0	1

To have the problem, as expressed by Billy Wadsworth, suggests that it is in part due to rounding errors — a change in order of the rows and columns might then solve the problem by not having perhaps so many rounding errors.

To accept a pseudo-inverse is up to the individual, but, in case, here is an algorithm for the Cholesky pseudo-inverse routine. It is worth noting that the two-dimensional array A(n,n) can be simulated by a one-dimensional array

$$B(n(n+1)/2)$$

in the case of a symmetric matrix. The cell A(j,i) is then the equivalent of the cell

$$B(i(2n+1-i)/2-n+j)$$

I have written an Atom routine on this basis. Only the lower-triangular part of the matrix is used in the routine, so I give the example matrix and after each stage the new contents of the matrix so that you can check the translated algorithm.

Input matrix			
	4	5	2
	5	6.5	1.5
	2	1.5	6

(continued on next page)

Table 1. Comparison of Sharp PC-1211 and Casio 702-P



	Sharp PC-1211/ Tandy	Casio702-P
Price	£79.90 from Compshop-London	£119.95 from Tempus-Cambridge
Keyboard	QWERTY 57 keys	ABCD 65 keys
Display	24 character seven-by-five dot matrix. 10-digit mantissa	20 characters of similar type
Memory	Non-volatile User-definable: From 1,424 program steps/26 memories to 208 program steps/ 178 memories All standard scientific functions except hyperbolic and statistical	Non-volatile User-definable: From 1,680 program steps/26 memories to 80 program steps/226 memories All scientific functions including linear regression and standard deviation
Arithmetic	15 levels of parenthesis priority as in standard Basic Maximum subroutine nesting = 4	20 levels of parenthesis and priority as in standard arithmetic Maximum subroutine nest = 10
Accessories	CE-121 cassette interface CE-122 printer	FA-2 cassette interface FP-10 printer and ROM packs are available next year

```
   1    print n;"<";a;b;c;d;e;f;"(mine)"
   2    print m; l;k;j;i;h;">";g;"(yours)"
   3    return
   5    t=0 : for α=8to13: t=t+(a(α)<>0): next α
   6    return
  10    "k" clear: pause"Kalah program by R.terry":pause" 5 stones/pit
 to start."
  15    for α = 1 to 6 : a(α) = 5: a (α+7)=5 :next α: gosub 1
  17    pause "when you cant move,"pause" enter 0 when asked":pause
"<what pit?>"
  20    input "0=i start, 1 = you start";s :ifs goto 115
  25    beep 1:pause "my go"
  30    u=0:for α=1 to 6:a(α+26)=1
  32    if a (α)=0 goto 65
  35    if a (α)>u let u=a(α)
  40    w=α-a(α) : w=14*(w<1)+w : w=w-(w=7): p=α+26
  42    if w=14 let a (p)=2.5 : goto 65
  45    a(p)=a(p)+(a(α)=u)
  48    a(p)=a(p)+(a(α+1)=1)*(α<6)
  52    a(p)=a(p)+(a(w)=0)*(w>7)+(a(w+1)=1)*(w>13)
  60    a(p)=a(p)+(a(w) =0)*(w<7)*(a(14-w)>0)
  65    next α : u=1
  70    for α = 6 to 1 step - 1 : if a (α+26)>u let u=a(α+26):r=α
  75    next α: beep r: if u=0 pause " i cant move ": goto 165
  80    print "i empty pit";r:w=r-1
  85    for α=1 to a (r) :w=14*(w<1)+w:w=w-(w=7)
  90    a(w)=a(w)+1:w=w-1:next α: a (r) =0: w=w+1
 105    if (a(w)=1)*(w<7)letn=n+a(14-w)+1: a(14-w)=0:a(w)=0
 110    gosub 1: if w=14 pause "my go again" : goto 30
 115    beep 1: pause "your go"
 120    input "what pit? (1-6)";r:r=14-r : if (r<8)+(r>13) goto 160
 125    if a(r)=0 goto 120
 130    w=r-1 : for α=1 to a (r) :w=14*(w<1)+w:w=w-(w=14)
 135    a(w)=a(w)+1: w=w-1: next α: a(r) = 0
 145    if (a(w+1)=1)*(w>6) let g=g+a(13-w)+1: a(13-w)=0:a(w+1)=0
 150    gosub 1: if w=6 pause"your go again":goto 120
 155    goto 25
 160    gosub 5: if t print "cheat! you can move":goto 120
 165    if g>n print "you win...this time...":end
 170    if g<n print "i win (ha! ha!): end
 175    print " its a draw.....": end.

 type shift K in DEF mode to run
```

Listing for Kalah or Owani.

(continued from previous page)

Cholesky Decomposition

E = 1.0 E−5 ; Criterion for pivoting
For J = 1 to N ; Go down each column starting at the diagonal
If J = 1 goto L1 ; The first column is slightly odd
For K = 1 to J−1 ; If J = 1 then J−1 is zero, so it is odd
A(J,J) = A(J,J) − A(J,K) × A(J,K) ; Much better than using a power of two
EndloopK

L1: If A(J,J) less than E then
A(J,J) = 0 ; Pivot check, for sqrt of negative
If A(J,J) more than E then
A(J,J) = 1/sqrt(A(J,J)) ; Normal action
If J = N goto L2 ; Everything has finished
For I = J + 1 to N ; Now we are on to the off-diagonal
If J = 1 goto L3 ; Jump over this K loop
For K = 1 to J−1 ; If J = 1 then J−1 is zero
A(I,J) = A(I,J) − A(I,K) × A(J,K)
EndloopK
L3: A(I,J) = A(I,J) × A(J,J) ; Do not divide, as reciprocal already
EndloopI
L2: EndloopJ

Matrix			
	.5	2.5	1
	2.5	2	−2
	1	−2	1

Inverse Cholesky Matrix

For I = I to N−1 ; Columnwise from 1 to N−1
ForJ = I + 1 to N ; The diagonal element is OK
X = 0 ; Temporary storage
For K = I to J−1 ; Collects a row and column multiple
X = X + A(K,I) × A(J,K)
EndloopK
A(J,I) = −X × A(J,J) ; The final move
EndloopJ
EndloopI

Matrix			
	.5	−2.5	−5.5
	−2.5	2	4
	−5.5	4	1

Forming Inverse Matrix

For I = 1 to N ; Column-wise
For J = I to N ; From diagonal down
X = 0 ; Temporary
For K = J to N ; Row and column multiple
X = X + A(K,J) × A(K,I)
EndloopK
A(J,I) = X ; That is it
EndloopJ
EndloopI

Matrix			
	36.75	−27	−5.5
	−27	20	4
	−5.5	4	1

If you found that you could not quite lift off with the flight simulator program in the October issue, it may have been due to lines 46 and 57 which should have read X≠0?, and lines 171 and 296 X≠Y?

This month's finisher is another small brain-teaser. Consider a row of four points:

● ● ● ●

If you are at an end then the only way of moving is back towards the centre; at the middle points the probability of going to the left or right is exactly equal to 1/2. Starting at one end, what is the expected — or average — number of moves to take you to the other end?

The solution is asymptotic, so take two consecutive answers separated by less than 10^{-6} to be the ending criterion. A year's subscription to the most elegant solution. ■

4th FORTH for TANGERINE

FORTH - the dictionary based language which is generating so much interest is NOW available to Tangerine users. FORTH Is both an interpreter and compiler offering the power of a high level language with the flexibility and speed of assembler.

Supplied on cassette is:

- A full standard implementation of fig-FORTH release 1.1 (variable length names and compiler security) with RAM simulation of disc
- Plus in-line assembly via linkage to the XBUG line assembler
- Plus a FORTH editor
- Plus a cut down version offering 8K users (Micron or Microtan + Tanex) increased dictionary space.

Price — £24.95 inclusive of VAT, p&p and documentation.

Documentation consists of the FORTH model overview, the FORTH glossary (containing definitions of all FORTH words), editor instructions and implementation instructions. A FORTH primer or user manual is recommended for newcomers. The full implementation ideally requires 12 to 16K of RAM, although it will load into 8K (but then dictionary space is limited and lack of buffer space precludes the use of input/output). The cut down version, by excluding i/o and related utilities, offers the 8K users extra dictionary space.

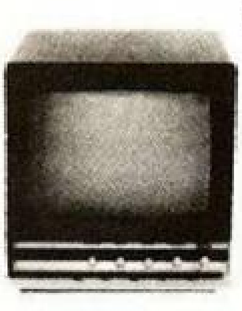
For those wishing to customise their own versions, assembler source code can also be supplied for an extra cost of £5. This is suitable for assembly by the Microtanic 2 pass assembler. EPROM implementations also undertaken by request.

Orders (sent by return where possible) to:

MICROTANIC SOFTWARE,
235 Friern Road,
Dulwich, London SE22

4th 4th

The *SENSATIONAL* CROFTON Offer



9" metal cased monitor at lowest ever price — £48.50 plus VAT — £55.77 total plus carriage.

P31 green tube models available at **£60** + VAT — **£69** total + carriage

STOCKISTS IN ALL AREAS WANTED. WRITE FOR DETAILS.

12" Model available at following prices.
Standard P4 white **£145** + VAT
£166 total + carriage
Standard P31 green **£160** + VAT
— **£184** + carriage

All major credit cards accepted. Ask for Crofton Mail **Order Catalogue.**

All items subject to availability. The above prices include VAT
Carriage will be charged at cost

Phone or write to

CROFTON ELECTRONICS LIMITED
35 Grosvenor Road, Twickenham, Middx TW1 4AD.
Tel: 01-891 1923/1513

CASIO
WORLD BEATERS
World's Most Powerful BASIC Pocket Computer

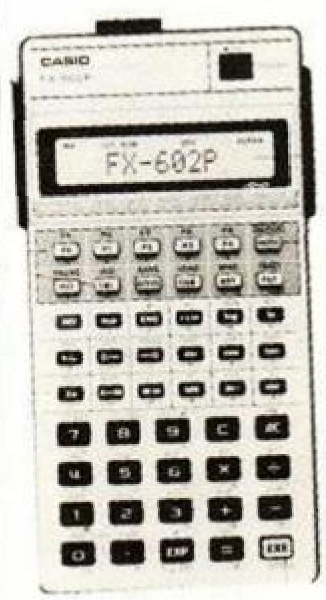


FX-702P RRP £134.95 **ONLY £119.95**

Flattens the Sharp PC1211

Alpha/numeric dot matrix scrolling LCD. Variable input from 1680 steps, 26 memories, to 80 steps, 226 memories, all retained when switched off. Up to 10 programs. Subroutines; 10 levels. FOR: NEXT looping; 8 levels. Debugging by tracing. Editing by moving cursor. 55 built-in functions, including Regression Analysis and Correlation, all usable in programs. Program/data storage on cassette tape via optional FA-2 adaptor, £19.95. Two lithium batteries give approx 200 hours continuous operation, with Auto Power Off.
17 x 165 x 82mm (5/8 x 6½ x 3¼") 176g (6.2oz).

FREE MICROL PROFESSIONAL PROGRAMMING PACK
supplied free of charge with every FX-702P or FX-602P purchased from us.
Only on request, at time of ordering. RRP £9.95



World's Fastest Programmable?
FX-602P

- LCD alpha/numeric (dot matrix) scrolling display (86 types).
- Variable input from 32 program steps with 88 memories, to 512 steps with 22 memories.
- Memory and program retention when switched off.
- Up to 10 pairs unconditioned jumps (6010). Manual jump.
- Conditional jumps and count jumps. Indirect addressing.
- Up to 9 subroutines, 9 levels.
- 50 built-in scientific functions, all usable in programmes.
- PAM (Perfect Algebraic Method) with 33 brackets at 11 levels.
- Program storage on cassette tape using optional FA-1, £19.95
- Campatible with FX-501/2P.
- 2 lithium batteries. Approx 660 hours use with Auto Power Off.
- 9.6 x 71 x 141.2mm. 100g.

ONLY £74.95 (RRP £84.95)

FP-10 MINI PRINTER
For FX-702P, FX-602P, FX-601P, FX-502P, FX-501P
Available soon. Price and delivery on application.

World's Most Versatile
Alarm Chronograph Watch
AX-210



10 alternative displays; over 60 functions. LCD ANALOG display of time, plus: DIGITAL display of: Time (12 or 24 hour); Calendar; Full month calendar (this month and next month); Dual time (12 or 24 hour); Alarm time; Countdown alarm timer with memory function; Professional 1/100 second stopwatch with laps, etc. Hourly time signal. Alarm — electronic buzzer or 3 selectable melodies. Rapid forward/ backward setting. 9.4 x 35.4 x 36mm.

(RRP £34.95) **ONLY £29.95**

SHORT FORM CATALOGUE
Available on request. 14p stamp appreciated

Price includes VAT, P&P. Delivery normally by return of post.
Send cheques, P.O., of 'phone your Access of B'card number to:-

TEMPUS

Dept WW/12 FREEPOST, 164-167 East Road, Cambridge CB1 1DB
Tel: 0223 312866 YC12

ZX81 + 16K — STELLAR SUPER NEW SOFTWARE

SPACE BATTLE

Using superb dynamic flicker-free graphics, this 'Arcade' type game with continuous score display is instantly addictive. Destroying the Xylon fleet calls for swift reactions. Xylon battleships can deflect your photon torpedoes; you must escape into hyperspace to avoid destruction. If you succeed, the action resumes. Can you beat the highest score?

S.A.S. MISSION IMPOSSIBLE

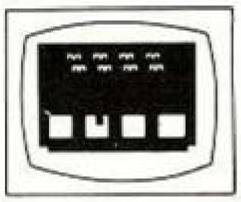
As a member of the crack S.A.S. hit team, your mission is to rescue ten hostages hidden in an embassy complex and kill their terrorist captors. Bombs explode about you as you blast your way through to the diplomats. Can you escape safely with all the hostages. This exciting game calls for cunning and daring to succeed. No two missions are alike!

Both games available on one cassette for £4.95 inc. p&p
Make cheques payable to:

STELLAR SOFTWARE
144 Pampisford Rd., South Croydon, Surrey CR2 6DA

MACRONICS

ZX81 26 Spiers Close, Knowle, Solihull, West Midlands, B93 9ES England **ZX80**



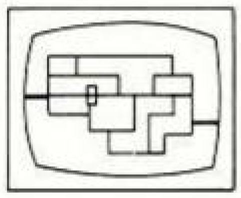
1K Space Intruders £5.00
3K Super Version £6.00

Written in machine code to give fast moving 'Flicker Free' graphics. A superb achievement.
Supplied on cassette with listing



16K High Resolution £4.00
(192 x 184) graphic pictures

Never before achieved on the ZX. Create your own artwork.
Supplied on cassette with listing and demonstration picture



9K Nightmare Park £3.75

The park of DEATH — can you get through this nightmare. No human has yet succeeded. Confronted by all sorts of 'Happenings' you'll be taken to your wits end.
Supplied on cassette only

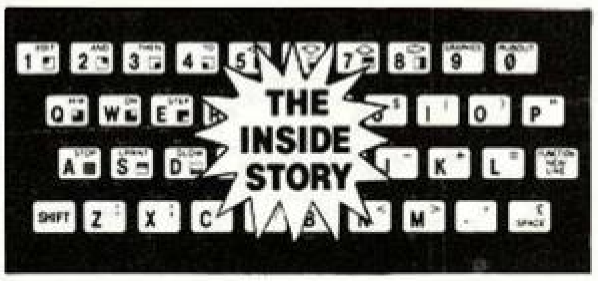
Specify 4K, 8K ROM or ZX81 when ordering.

NEW INTERACTIVE GRAPHICS for the 16k ZX81

DRAGON MAZE... are you cunning enough to evade the dragon? you are impeded by darkness: the walls are only displayed when you make a bad move, this gives the dragon an advantage. You have to play by the rules - the dragon breaks them when he gets angry. Fun for all ages!
Cassette and listing.......... £5.00 *(for ZX 81 only)*

"OUTSTANDINGLY USEFUL"

THE ZX81 COMPANION



Bob Maunder

LINSAC

Price £7.95 incl. UK postage
ISBN 0 907211 01 1

THE ZX81 COMPANION was reviewed in the September 1981 issue of the Educational ZX80/81 Users' Group Newsletter as follows:

Bob Maunder's ZX80 Companion was rightly recognised to be one of the best books published on progressive use of Sinclair's first micro. This is likely to gain a similar reputation. In its 130 pages, its author does not go as far as he did before, but his attempt to show meaningful uses of the machine is brilliantly successful.

The book has four sections, with the author exploring in turn interactive graphics (gaming), information retrieval, educational computing, and the ZX81 monitor. In each case the exploration is thoughtfully written, detailed, and illustrated with meaningful programs. The educational section is the same — Bob Maunder is a teacher — and here we find sensible ideas, tips, warnings and programs too. The monitor listing (0000 to 0CB9), while unique, is less fully backed up, and will be of no use to the ZX81 beginner without some knowledge of Z-80 assembly.

To conclude — this book is definitely an outstandingly useful second step for the ZX81 user.

Send cheques for £7.95 to:

LINSAC (YC)

68 BARKER ROAD, LINTHORPE,
MIDDLESBROUGH TS5 5ES

The gateway guide to the ZX-81 and ZX-80

By Mark Charlton. Published by Database Consultancy, 180 pages, paperback. Price £5.95.

ON THE WHOLE, this book, unsurprisingly, follows the standard Hartnell/Database pattern. It is a collection of programs, mainly games, with notes on ZX programming along the way. That is a well-tried and, without doubt, very successful pattern — even if it is one which does not add much sparkle to life.

This particular follower of the pattern is no better and no worse than the others. If you have just started with the ZX-81, it will teach you no more and no less than the others. If you are beyond the basics, it will give you more ideas for simple games.

Charlton's book has a few pluses — better binding, somewhat better layout and accuracy and some useful appendices. It does not, however, tackle anything serious, there are plenty of examples of poor programming, and there is no index.

Conclusions

- If you have not advanced very far into Sinclair microcomputing, this book is as good as any other introductory collection of programs for these machines.
- If you already have plenty of material, it will not offer much apart from a few more game variations.

Eric Deeson.

Atomic theory and practice

Published by Acorn Computers. Supplied with each Atom or supplied separately for £8.00.

TO A NEWCOMER to computing, there is one component of a computer system which is of crucial importance — the instruction book. Acorn seems to be aware of this and the book supplied with the Atom is of a particularly high quality.

It tackles the subject with a delicacy of touch rarely found in a technical manual. The inexperienced user is shown that programming a computer is reasonably straightforward.

At the same time, the more experienced reader is advised to turn to an alternative section which corresponds better to his interests and experience. The book also deals with assembly-language programming, and to make it easier to find the chapter required, the book has colour-coded sections.

Most features are illustrated with excellent program examples which reinforce the text. There are additional reference sections, covering just about everything you need to know.

Many manuals cover their machines as comprehensively as the Atom one, but few are as well designed to help you quickly find the information you require. An exceptional 11-page index comes to your assistance if contents page and colour coding fail.

Conclusions

- Atom Theory and Practice is the hand-book against which other manuals should be judged.

Alan Taylor

Atom Business

Published by Phipps Associates, 3 Downs Avenue, Epsom, Surrey. Price £6.95.

THIS BOOK is most interesting and unusual. As the title suggests, it contains business programs and the most notable ones are the nominal ledger system and the pair to keep sales records and plot the results graphically.

The book would be of most use to someone with a fully-expanded Atom and a printer, but there are items of interest to most Atom users. For example, in the sales graph, a technique is shown for mixing text and high-resolution graphics. Additionally, ways are shown to save large amounts of data to cassette and read them as required.

The programs for the most part are menu-driven and easy to load and use. They do their jobs very well and the book includes some useful tips and explanations.

One particularly noteworthy feature is the way that each program has its own chapter, divided into three parts. The first part explains what the program does, the second explains how to operate the program — usually including a sample run — and the third part gives the listing and explains how the program works.

Conclusions

- This book is particularly appealing as it shows how the Atom can do useful jobs rather than just play games.
- Very few books show how to use popular computers in serious ways, and Atom users are lucky to have a work of this quality at their disposal.

Alan Taylor.

Video/Computers

By Sippl and Dahl. Published by Prentice Hall International. Price £5.55.

THIS IS A frustrating and eventually worthless book. The full message on the front cover is: "How to select, mix, and operate personal computers and home video systems".

The concept that information technology is spreading into people's homes and that it is desirable for the hardware to converge into one grand information terminal is faultless; it is only the book itself which is flawed. There are many illustrations in the book, all of a typically American high standard, but the captions appear to be taken directly from the manufacturer's advertising material. Why do you need nine pictures of colour TV cameras one after another for any purpose apart from filling space?

The book fails to carry through its theme and there is no serious discussion of how a system might be assembled and no mention of the major problems of incompatibility both at an electronic-signal level and at a data-transmission protocol level.

You will be interested to know that the Sinclair ZX-80 is described thus: "Despite these minikin proportions, it does everything that larger, more expensive home computers do".

There are factual inaccuracies as soon as the book drifts away from the material supplied by the makers of a host of hardware.

Conclusions

- If you want your imagination stimulated and fuelled with accurate, up-to-date information on the silicon revolution, this is not the book to do it.
- I regret the loss of the trees which were pulped and wasted for the printing of *Video/Computers*.

John Dawson

Mastering machine code on your ZX-81 or ZX-80

By Tony Baker. Published by Database Consultancy, 180 pages, paperback. Price £5.95.

WHILE IT is possible to prepare good ZX programs in Basic, it is not possible to do so with full efficiency, high speed and high effectiveness. This book is an exceedingly brave attempt to introduce the thousands of ZX novices to the effective use of machine code. It consists of 180 close-packed pages of text and routines through which the author adeptly leads the diligent reader from first steps to a complete understanding of the subject.

The reader must be diligent, though. There is little point in skipping through until one finds some exciting routine and tries to enter it.

One must work at machine code, step by step. One must also be diligent to overcome the few errors in the book's first real machine-code program.

To use this book properly — and by the end of it be able to write long, useful programs and routines — you need to give yourself a solid week in the Common Cold Research Laboratory. Yet even after three or four hours — and with the benefit of some background knowledge — I can now work through published programs and see what they are intended to achieve.

In those 180 pages we find all kinds of techniques which should be used in commercial cassettes. The main theme is a 1K draughts program. There is also material on character generation, keyboard scanning, making music, displaying a series of pictures at speed, disassembly and arithmetic. There are games and serious routines, and the final program is a delight.

Conclusions

- A beautifully-structured guide for the uninitiated which pulls no punches and yet which is easy to use.
- It contains plenty of useful machine-code routines and programs.
- If you can tolerate the poor presentation and tiny type — the end result is a new world.

Eric Deeson

Getting acquainted with your Acorn Atom

Published by Database Consultancy, 105 Fairholme Avenue, Gidea Park, Romford, Essex. Price £7.95.

THIS WORK has fallen into the trap which has claimed so many victims among computer volumes. Books written as complete programming courses, in which you follow examples to master the computer, always duplicate material contained in the manuals.

The style of this book is to give a program listing in sections, with explanatory comments about what is happening. A cynic might suggest that this is a way of making 4oz. of programs fit a 2lb. book, but the comments may be of great interest to some readers.

My greatest reservation is that the examples given have a large number of errors in them. Most are of a minor nature, but in a book which assumes negligible experience on the part of the reader, such errors should not be present.

Many of the programs were originally written for other machines and have been converted. This means they do not take advantage of the Atom's programming strong points. We are even told to use the Stop command — the Atom does not have one.

Conclusions

- There are some good things to be said for the book; it gives you many ideas to try, and the style may be just what some people need to help them on their way to writing their own programs.
- Perhaps the best advice is to look at this book in a shop and see for yourself if it is what you need.

Alan Taylor.

UK 101 — OHIO — SHARP
TRS 80 — V.G. — ZX80/ 81

MICROTRAIN
At last you can combine two of the most popular hobbies in the country — railways and computing. MICROTRAIN, a brand-new quality game simulation from Premier, allows you set up a network of tracks, points, stations, tunnels, bridges etc. and then run trains to your own timetable! Signalling is provided either automatically or to your design.

MICROTRAIN is a screen-based simulation. With one keystroke you can draw and signal lines many scale-miles long. Up to five trains can be run simultaneously and the speed and length of each train is user-selectable. A cursor is used to move the whole screen display left or right for visual inspection of other parts of the network. The size of the network is only limited by your RAM or your imagination. You can use any of the machine's graphics on-screen to build up scenery, etc. Designs can be stored on to tape/disc for future retrieval.

If you already own a model railway, MICROTRAIN will allow you to transfer your design on to computer and run the two together. MICROTRAIN has been written by the writer of WORD4WORD and TOOLKIT2, a sure guarantee of quality. It is entirely in machine code and will run on any Video Genie or TRS80 (Model I or III). State machine and cassette or disc when ordering.
CASSETTE £14.95 inc. DISC £16.95 inc.

WORD4WORD
Stunning new TRS80/ VG WORD PROCESSOR
Why pay £75 to £150+ for a W.P.? W4W gives you all the facilities you could wish for in WP for a sensible price. Amongst the features included are:
FULL SCREEN EDITING & TWO WAY SCROLLING
GLOBAL TEXT INSERT/DELETE ON SCREEN
GLOBAL SEARCH & AMEND
TEXT SEARCH — TOTAL TEXT MOBILITY
TEXT FORMATTING TO SCREEN OR PRINTER
CASSETTE/DISC STORAGE OF TEXT
VISION LOAD OF STORED TEXT
STANDARD LETTER ROUTINES
OVERTYPE CORRECTION
SINGLE KEY WORD DELETION
PAGE LAYOUT & NUMBERING
TOTAL PRINTER CONTROL
TEXT HIGHLIGHTING FOR U/L RECOGNITION
TEXT BLOCK MANIPULATION
Many other features are included in W4W. We would need several pages to do justice to this superb product. W4W is comparable to most purpose built systems. Commissioned and developed by Premier with the hobbyist and small businessman in mind.
Please state machine type and printer when ordering.
CASSETTE £33.95. DISC £37.95

TOOLKIT 2 for TRS80 I & III/ VG
TOOLKIT 2 now features 17 new easy-to-remember command words which greatly enhance an already powerful BASIC and gives you a machine-code monitor for the first time.
REPLACE — find and replace any string, word or variable in a BASIC listing.
VARS — gives a list of variables on screen.
TRACE — displays the WHOLE line of BASIC during execution. User selectable speeds.
ABBREV — 26 BASIC words become single-key entries.
RENUMBER — operates from any start in any increment.
BLANK — removes unwanted spaces and LET statements.
VTAPE — gives true Vision loading of incoming program, plus APPENDing.
MC — a full machine code monitor.
VARTRACE — lists lines as executed plus variables.
PAGE — controlled list scrolling.
OLD — retrieves lost programs!
FIND — anything in a BASIC listing.
REMKIL — kills REMS!
DUPL — copy existing line to new line.
LFIND LVARS LREPLACE — all work to printer.
CASSETTE £29.95. DISC £32.95.

CHRISTMAS GAMES PACK
Available for the UK101, Superboard, TRS80, Video Genie, SHARP and Microtan, PREMIER's Christmas Games Pack will entertain youngsters and challenge adults, Super Santa, Reindeer Roundup and Toboggan Run. Three game pack for only £7.95

TOOLKIT 2 for UK101/ OHIO
The most powerful TOOLKIT on the market, TOOLKIT 2 gives you all the following facilities in only ONE EPROM.
REPL — exceptionally powerful Global Search and Replace of BASIC listings.
DUPL — copy a line into a new line.
LIST/ — controlled listing of program
FIND — anything in a BASIC listing
RENUM — renumber from any start in any increment — full error messages, totally reliable
AUTO — generate new line numbers automatically, any start, any increment
DELETE — high-speed block line delete
VIEW — examine cassette contents without loading to memory
TRACE — superb trace feature — screen transparent. Can be turned on and off within a program
MC — enter the monitor quickly!
TOOLKIT 2 also lists the relevant line of BASIC where any error occurs and cures the warm start 'OM ERROR' bug
Available in EPROM only (8000hex), for CEGMON, MON01 and 2, and SYNMON monitors (DISK soon), Price £19.95. State machine and monitor when ordering.

BASIC 5 for UK101 and OHIO
The most devastating enhancement yet, adding 17 new BASIC words to your interpreter which can be used in program lines and give machine-code response speed to graphics and formatting.
HLIN, VLIN, SCR, BLK, SET and TEST allow generation and manipulation of graphics at speeds which are unobtainable in BASIC.
PRINTUSING, PRINTAT, INAT allow total control of screen input/output.
GET (key), RD (Read DATA), GS and GT (GOSUB and GOTO a variable), GO and GO$ (GOTO a machine code routine), allow total program flexibility.
WI and CWI allow CEGMON users to manipulate their screen under variable control, using one command, in Hex or Decimal.
BASIC 5 is available for CEGMON and MON02 only. State precisely your computer and monitor when ordering. Comes complete with comprehensive manual.
Available on DISC or in EPROM (9000hex) £19.95.

BASIC 4 cassette file handling system
This new EPROM for the UK101/OHIO provides a comprehensive file-handling system, capable of working at up to 4800 baud.
* Named programs to cassette
* Verify tape contents facility
* Reliable high speed save/load
* Selectable auto-run of loaded BASIC program
* Crash recovery command (OLD)
* Original SAVE/LOAD commands unaltered
* Reduces LOAD/SAVE times
* Seven new SAVE/LOAD commands
* Non-destructive memory test
* Initialises BASIC 5 automatically if resident

BASIC 4 is a plug-in replacement for your existingf BASIC 4 ROM.
Price £11.95

SOUND/ V.I.A. BOARD
The TES II VIA/SOUND kit gives you up to 56 Input/Output lines and programmable sound generation. In order to allow you total flexibility in designing your system, we are offering the kit in low-cost packs.

The Base Kit consists of PCB, connector, address decoding and buffering, plus IC sockets.

The Sound Pack consists of AY-3-8910 sound chip, amplifier and components.

The VIA Pack consists of VIA and support.
BASE KIT £24.95. SOUND £11.95. VIA £9.95.

TES II HARDWARE RANGE

PRODUCT	KIT	BUILT
8K RAM BOARD	£29.95	£39.95
8K EPROM BOARD	·29.95	£39.95
8 Slot MOTHERBOARD + PSU	£29.95	£39.95
JI BUFFER BOARD	£19.95	£29.95
MINI EPROM/ROM BOARD	£14.95	£20.95
SCREEN ENHANCEMENT KIT	£55.95	£69.95

SPECIAL OFFERS

TOOLKIT 2 + MINI EPROM BOARD	£29.95
BASIC 5 + MINI EPROM BOARD	£29.95
CODEKIT + MINI EPROM BOARD	£29.95
SOUND/VIA — Base, Sound and VIA kits	£43.95

POSTAGE & PACKING
Software 60p per order, EPROMS/DISCS 80p per order,
HARDWARE £1.50 per item. Maximum £3.00
ALL PRICES INCLUDE V.A.T.

BARCLAYCARD VISA **Premier Publications** We welcome Access

12 Kingscote Road Addiscombe Croydon Surrey Telephone 01-656 6156

SOFTWARE FILE

Software File gives you the opportunity to have your programs, ideas or discoveries published. We will accept contributions for any personal computer and will group programs for like machines together in the file. Please double-check your listings before sending them. Mark your letter clearly for *Your Computer.* We will pay £6 for each contribution published.

Machine-code tips

W E Thomson, Aldeburgh, Suffolk.

ZX-81

TREVOR SHARPLES' article in the August/September issue did not perhaps draw a sufficiently sharp distinction between working with the ZX-80 and ZX-81.

For example, in talking about putting machine-code programs in Rem statements, he said:

> Do not list the Rem statement. Sometimes, but not always, this will cause the ZX-80/81 to crash.

True enough, but the statement hides the vast difference between the machines. One-quarter of the character codes — those from 64 to 127 — appearing in a Rem statement listed by the ZX-80 cause chaos. The ZX-81 is much more placid: only 118 and 126 bother it.

They lead to corruption of the listing in degrees of severity varying from the omission of six characters within the REM, to a listing that stops half-way through the Rem. There is, however, no crash; only the listing is corrupted and the program can be run without trouble.

The other example is the renumbering routine. The text has:

> The ZX-80/81 will then execute line 9070.

This gives the impression that the routine can be used for either machine, whereas it can be used only for the ZX-80. This routine advances from line to line by looking for character 118 — Newline — knowing that, apart from the line number itself, 118 can appear nowhere else in a statement.

In the ZX-81, however, 118 can also appear in the bytes the listing does not print — namely, those that give the text length and those that give the binary form of numerical constants. A ZX-81 renumbering program must use a different approach.

My program uses the text length to skip to the end of each line. It also detects the end of program by looking for the address where the display file starts, so there is no need to have a 9999 line in the program.

A possible loading program is:

```
1000  LET A=(address of
1st byte of program)
1001  FOR I=A TO A+30
1002  INPUT B
1003  PRINT B,
1004  POKE I,E
1005  NEXT I
1006  CLS
1007  RAND USR A
1008  LIST
```

This differs from that associated with the ZX-80 renumbering program but the differences are not vital — just a matter of individual preference.

The program renumbers in steps of 10, but only one byte need be changed to alter this. Instead of 10 — about half-way down — use whatever you want. Or, if the routine has been loaded, Poke A+17,n will make the change.

It may be as well to remind you that such a renumbering program does only half the job. The arithmetic expressions following Gotos must be altered by editing to agree with the new numbering scheme, with possibly other changes.

Bytes in decimal		Symbolic Instr	Comments
33 124 64	start:	LD HL, 16508	HL=ProgramStart -1
17 0 0		LD DE, 0	Newlinenumber =0
35	next:	INC HL	HL=address of 1st byte
237 75 12 64		LD BC, (D FILE)	If
167		AND A	HL=D FILE
237 66		SBC HL,BC	then
200		RET Z	return
9		ADD HL,BC	else
6 10		LD B, 10	Newlinenumber=
19	inc:	INC DE	Newlinenumber+
16 253		DJNZ inc	10:
114		LD(HL),D	Linenumber
35		INC HL	=
115		LD(HL),E	Newlinenumber:
35		INC HL	HL=
78		LD C, (HL)	HL+
35		INC HL	Textlength
70		LD B, (HL)	(leaving HL set
9		ADD HL,BC	to last byte of line)
24 231		JR next	next line

Security numbers

A D Robinson, Wath on Dearne, South Yorkshire.

ZX-81

WITH THE popularity of private bank-account and home-finance programs for the ZX-81, my routine requires the entry of a security number before a program will run. Any incorrect entry will mean the execution of New, and the program will vanish.

Because the routine is written in machine code, it is possible to break out or return to Basic without the correct code which can be made up of any number of digits in length.

The first line of the program is a Rem statement into which the machine code will be Poked. The number of Xs in it is determined as follows: 10 for each digit of the security code plus 21. For example a three-digit security code would need 51 Xs in line 1. Enter this short program before your main program:

```
1 REM XXXXXX......etc.
2 LET A=16514
3 INPUT B
4 POKE A,B
5 LET A=A+1
6 GOTO 3
```

Now run this, and enter the following numbers, pressing newline after each:
205,187,2,44,32,250,205,187,2,68,77,81,20,40, 247,205,189,7,126,201.

The next few numbers form the machine code to check one digit. Enter them as required depending on the number of digits in your chosen security code.

205,130,64,254,44,40,3,205,0,0.

The number underlined is the one you must alter to the code of any key you choose — at the moment it is set for G. Finally enter 201, and then press to halt the program. When you

(continued on next page)

(continued from previous page)

List, you will see that the Rem statement now contains the machine code. Lines 2 to 6 can now be deleted — their job is done. Now enter:

```
2 RAND USR 16534
```

To make the program Run automatically after loading, enter the following at the end of your main program:

```
9995 STOP
9996 SAVE "ANYNAME"
9997 GOTO 2
```

Save the whole program by typing Goto 9996. It will save as normal, but, more importantly, when loaded back from tape it will run immediately. This means that no-one can gain access to the main program without the exact code.

The program works by scanning the keyboard waiting for a key to be depressed. When this occurs, a routine is called in the ROM which finds the code of that key. This is then compared with the code you have programmed. If it is identical, the program jumps to the next section to wait for another key depression; but if it is different, the program calls the routine in ROM which executes New.

As any ordinary key or shifted key code can be selected, even a two-digit security code gives more than 6,000 possible combinations.

Card Sharp

C J Davison, Newton Abbot, Devon.

MZ-80K

MY ROUTINE, creates, stores and shuffles a pack of cards in the normal way. The advantages of this are that the shuffled array may be called in order — i.e., A$(1) to A$(52) — and yet the output is completely random. Once you have gone through all the cards, you just call the shuffle routine, and you are ready to go.

Using this structure, Pontoon for instance becomes very simple and easy to write since the cards do not need to be picked randomly and the whole pack can be used. The Basic used is Xtal 2.2. However, because of its simplicity and size — less than .5K — it should work on any system without any alteration.

```
5       REM  ***  SET UP STRINGS ***
10      DIM A$ (53)
20      C=0 : S$ ="CLUBS..HEARTS.SPADES.DIAMONDS"
30      D$= "ACE..TWO..THREE FOUR.FIVE.SIX..SEVEN EIGHT NINE.
        TEN..JACK.QUEEN KING."
40      REM *** CREATE UNSHUFFLED PACK ***
50      FOR S = 1 TO 4
60      FOR N = 1 TO 13
70      C=C+1
80      A$(C)=MID$(D$,N*5-4,5)+".OF."+MID$(S$,S*7-6,7)
90      NEXT N
100     NEXT S
110     REM *** SHUFFLE PACK ***
120     FOR C= 53 TO 2 STEP - 1
130     A=INT(1+52*RND(8))
140     A$(C) = A$(A) : A$(A) = A$ (C-1)
150     NEXT C

FULL STOP IS EQUIVALENT TO A SPACE.
```

Real-time clock

Kamal Jabbour, Salford.

ATOM

HERE IS AN efficient way of realising a 24-hour, real-time clock for the Acorn Atom using its 6522 VIA. Once initialised, the program occupies only 130 bytes of machine code. Time is permanently displayed in hours, minutes and seconds at the top right-hand corner of the screen. The clock does not interfere in any way with the operation of the system. A programmable alarm facility is also provided.

Among the facilities provided by the 6522 VIA are two programmable counter timers, CT1 and CT2. Used in the free-running mode, CT1 — addresses # B805,6,7 in the Atom — creates an interrupt each time the count is decremented to zero. Note that link LK2 on the Atom must be closed for the interrupt request of the VIA to reach the 6502 microprocessor.

CT1 has 16 bits and is decremented at the Atom clock rate of 1MHz, so a programmable delay of up to 65,536μs. can be generated. In our application, CT1 is programmed to give an interrupt every 50ms. — line 100 in the program listing. The recurring interrupts are used as the time base for the clock.

The clock is incremented by one second every 20th interrupt — lines 250-260. Time is permanently stored at locations # A0-# A8 in ASCII format, and is copied on to the screen at top right-hand corner locations # 8017-# 801F at every interrupt to avoid any flicker of the time display during computer operation — lines 230-240.

Lines 270-360 increment seconds, minutes and hours as applicable, and reset the clock to zero at 2400 hours. Lines 365-400 compare the current time with the alarm setting stored in ASCII at locations # AB-# AE, every minute. The alarm can be disabled by storing zero in location # AF.

When the pre-set alarm time is reached, a user subroutine at L is called and executed — in my program L= # A000 where I have a utility EPROM. Lines 210 and 420 preserve the registers of the 6502 so that normal computer operation is not disturbed by the clock routine.

The Basic part of the program assembles the real-time routine and initialises the clock and the alarm. Press ESC if no alarm is required. After initialisation, only the 130 bytes of machine code are needed for the clock operation, and the source code can be destroyed.

The machine-code routine is relocatable by changing the value of P — P= # 2800 in this listing — and can happily reside in the utility EPROM. The alarm setting can be modified by changing the contents of # AB-# AE, e.g.:

```
!# AB = # 31373330
```

will set the alarm to 1730 hours. Note that the alarm is automatically disabled once reached. It can be enabled by storing a non-zero number in # AF.

Once the initialisation routine is run, the Atom can be used as normal. The clock routine spends some 50μs. at each interrupt to update the display. So the operation of the computer is slowed by about 0.1 percent only — insignificant, and programs run at normal speed.

The bell, CTRL-G, sounds a bit shaky as it is interrupted some 10 times. Obviously, the clock stops when loading or saving programs as the COS disables interrupts. The displayed time could always be updated by modifying the contents of # A0-# A7. Note that the interrupt is enabled when the Break key is pressed.

Finally, the clock may run slow or fast depending on the crystal in individual Atoms. This can be cured by changing the contents of # B805,6,7 — or line 100 — as the 50ms. delay is obtained by

$$(195)*256 + (80) = 50000$$

where 195 is the contents of locations # B805 and # B807, and 80 is the contents of # B806 for fine adjustment.

```
1     REM REAL - TIME CLOCK FOR THE ATOM, BY KAMAL JABBOUR
10    DIM BB5; GOS. a; GOS.a
20    IN. "TIME; HOURS"H, "MINUTES"M,"SECONDS"S
30    ?#A0=H/10+48;?#A1=H%10+48;?#A2=#3A;?#9F=32
40    ?#A3=M/10+48;?#A4=M%10+48;?#A5=#20
50    ?#A6=S/10+48;?#A7=S%10+48;?#A8=20
60    ?#AF=0
100   B=#B800;B?11=64;B?6=80;B?7=195;B?5=195;B?14=#C0
110   IN. "ALARM: HOURS"H, "MINUTES"M
120   ?#AE=H/10+48;?#AD=H%10+48
130   ?#AC=M/10+48;?#AB=M%10+48
140   ?#AF=1;E
180a P.#21
190   P=#2800;?#204=P%256;?#205=P/256;L=#A000
200[ LDA B+4
210   STX#A9; STY#AA
230   LDX@8
240:  BB2 LDA#9F,X; STA#8017,X;DEX;BPL BB2
250   DEC#A8; BNE BB3
260   LDA@20; STA#A8
270   LDX@#3A; LDY@#36; LDA@#30
280   INC#A7; CPX#A7; BNE BB3; STA#A7
290   INC#A6; CPY#A6; BNE BB3; STA#A6
300   INC#A4; CPX#A4; BNE BB4; STA#A4
```

(listing continued on next page)

(listing continued from previous page)

```
310  INC#A3; CPY#A3; BNE BB4; STA#A3
320  INC#A1; CPX#A1; BEQ BB5
330  LDX@#34; CPX#A1; BNE BB4
340  LDX@#32; CPX#A0; BNE BB4
350  LDX@#2F; STX#A0
360: BB5 STA#A1; INC#A0
365: BB4 LDA#AF; BEQ BB3
370  LDA#A4; CMP#AB; BNE BB3
380  LDA#A3; CMP#AC; BNE BB3
390  LDA#A1; CMP#AD; BNE BB3
400  LDA#A0; CMP#AE; BNE BB3
410  CLI; LDA@0;STA#AF; JSR L
420: BB3 LDX#A9; LDY#AA; PLA; RTI
430]
440  P.$6;R.
```

Demon at the wheel

Richard Bassett,
Sutton Coldfield,
West Midlands.

MZ-80K

DEMON DRIVER will run on a Sharp MZ-80K and uses approximately 2K. The idea of the game is to destroy as many people as possible without hitting any of the white barriers. The car moves across the screen and the key "B" will move it upwards and "M" downwards: "N" holds the car straight.

In a tight spot, the £ key will destroy all barriers and people in sight. The "V" key produces a machine-gun effect which can be used five times. After this, no firepower is produced until reloaded. Whenever you hit a person or barrier, a certain sound is produced. As well as the maximum number of people destroyed, the time is also given.

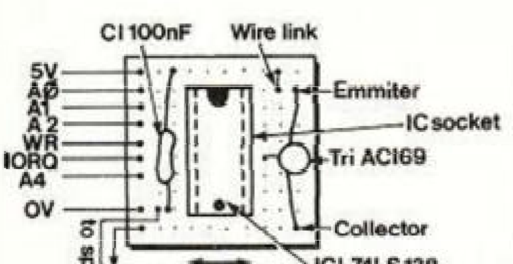
```
190 REM SET VARIABLES
200 TI$="000000":L=0:QW=0:UU=0:EE=0:AA=53748:WE=0
210 PRINT"■"
220 POKEAA,200
230 POKEAA-1,0
240 IF BB=1 THEN POKEAA+39,0
250 IF BB=2 THEN POKEAA-41,0
260 GET ZZ$
270 IF ZZ$="V" THEN GOSUB 470
280 IF ZZ$="B" THEN BB=1
290 IF ZZ$="£" THEN GOSUB 800
300 IF ZZ$="M" THEN BB=2
310 IF ZZ$="N" THEN BB=3
320 IF AA<53200 THEN BB=0
330 IF BB=1 THEN AA=AA-40
340 IF BB=2 THEN AA=AA+40
350 AA=AA+1
360 EF=INT(2*RND(1)+1)
370 IF EF=1 THEN GOTO 400
380 SS=INT(1000*RND(1)+1)
390 POKE53248+SS,67
400 UU=INT(1000*RND(1)+1)
410 POKE53248+UU,202
420 IF(PEEK(AA)=202)THENMUSIC"A1":L=L+1
430 IF (PEEK(AA)=67) THEN GOSUB 840
440 EE=EE+1:IF EE>200   THEN UU=0
450 IF EE>200  THEN EE=0
460 GOTO 220
470 UU=UU+1
480 IF UU>5 THEN RETURN
490 FOR FF=1TO10
500 POKEAA+FF,90
510 MUSIC"B1"
520 POKEAA+FF,0
530 NEXT FF
540 RETURN
550 REM EXPLOSION
552 POKEAA,255
554 FOR QQ=1TO300:NEXT QQ
555 POKEAA,0
556 IF Z1=1 THEN GOTO 640
557 POKEAA+1,250:POKEAA-1,245:POKEAA+40,252:POKEAA-40,243
558 POKEAA-41,249:POKEAA-39,246:POKEAA+39,246:POKEAA+41,249
560 FOR QQ=1TO300:NEXTQQ
562 POKEAA+1,0:POKEAA-1,0:POKEAA+40,0:POKEAA-40,0:POKEAA-41,0
564 POKEAA-39,0:POKEAA+41,0:POKEAA+39,0
565 IF Z1=1 THEN 552
566 POKEAA-80,243:POKEAA+80,252:POKEAA+2,250:POKEAA-2,245:POKEAA-81,246
568 POKEAA-82,249:POKEAA-79,249:POKEAA-78,246:POKEAA-42,246:POKEAA-38,249
570 POKEAA+42,246:POKEAA+82,249:POKEAA+81,246:POKEAA+79,249:POKEAA+78,246
572 POKEAA+38,249
574 FOR QQ=1TO300:NEXT QQ
576 POKEAA-80,0:POKEAA+80,0:POKEAA+2,0:POKEAA-2,0:POKEAA-81,0
578 POKEAA-82,0:POKEAA-79,0:POKEAA-78,0:POKEAA-42,0:POKEAA-38,0
580 POKEAA+42,0:POKEAA+82,0:POKEAA+81,0:POKEAA+79,0:POKEAA+78,0
582 POKEAA+38,0
590 Z1=1:GOTO 557
640 MUSIC"C3B3A3C2B2A2C1B1A1D6"
645 Z1=0
650 PRINT"■":POKE53740,25:POKE53741,15:POKE53742,21
660 FOR XX=1TO1000:NEXT XX
670 POKE53745,12:POKE53746,15:POKE53747,19:POKE53748,5
680 FOR XX=1 TO 1000:NEXT XX
690 XX=0
700 POKE53821,19:POKE53822,21:POKE53823,3:POKE53824,11:POKE53825,5
710 POKE53826,18
720 FOR XX=1TO1000:NEXTXX
730 XX=0
740 PRINT"■"
750 PRINT"YOU LASTED ";TI$:PRINT:PRINT"AND DESTROYED ";L;" PEOPLE"
760 PRINT:PRINT:PRINT"PUSH ANY KEY FOR ANOTHER GAME"
770 GET ZZ$
780 IF ZZ$="" THEN GOTO 770
790 GOTO 200
800 QW=QW+1
810 IF QW>2 THEN MUSIC"B2":RETURN
820 MUSIC"A1B1A1B1A1B1A1B1A1B1"
830 PRINT"■" :RETURN
840 WE=WE+1
850 MUSIC"C1"
860 IF WE>2 THEN 550
870 RETURN
```

Sound-box

ZX-81

J Chalmers,
East Grinstead, West Sussex.

HAVE YOU EVER wished that your ZX-81 could play tunes, give audible warnings, add sound effects to games or just make noises? Sound is made up of a series of clicks in rapid succession. If the ZX-81 can produce these clicks rapidly enough, the noise produced



Stripboard component layout

represents a musical tone. The more clicks per second produced, the higher the note.

To achieve a sufficiently fast speed, the sound box can be used only when the machine is in Fast mode. Again, for speed but also because of the limits of the Basic language, it is necessary to use a machine-code routine to produce the noise required. The note required is selected by Poke commands.

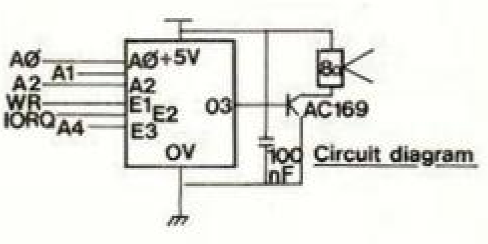
Poke 16526 will vary the frequency of the note and very fine tuning can be achieved. This Poke will from now be known as the fine-tune byte. Poke 16527 will vary the base frequency, and will be referred to as the coarse-tuning byte. The length of the note is set by Poke 16519, and this will be known as the length byte.

The circuit can be constructed on a piece of 10-by-10-hole 0.1in. stripboard. If the diagram is followed, construction should be simple. Only eight connections are needed to the computer and they can either be hard-wired, or taken to a 23-by-23-way edge connector plugged into the expansion port at the rear. The sound output is taken to a small 8-ohm loudspeaker. If a greater volume is required, the speaker can be disconnected and the output taken to a suitable amplifier.

The chip used in the circuit is a three-to-eight-line decoder. The three lines are the lowest three bits of the address bus, A0, A1, and A2. If the chip is enabled using all three enables lines provided and these lines are connected to A4, $\overline{IORQ}$, and $\overline{WR}$, it is possible to produce eight port-select lines for ports 8 to 15. By trial and error it was found that port 11 was not actually used by the ZX-81. This port-select output can, therefore, be used to make the speaker click every time the port is written to by the Z-80 Out(C), C instruction. This is where the machine code is required.

All you have to do is to write a short routine to call port 11, wait depending on the note required, and then jump back to calling the port. To make a note of certain length and then return to Basic is achieved by encasing the routine in another loop which returns to Basic after completing a number of cycles.

Listed here is a Basic program which enables the machine-code routine to be entered into the memory. Also listed here is a hex dump. After the Basic program has been started, you enter this hex code, two characters



Circuit diagram

at a time with a press of Newline between each pair.

When you have typed in the code a 9 error report will appear at the bottom of the screen; try entering Run 100. If you have done everything correctly your sound-box should be making a series of random notes.

When you have built your sound-box and made it work, you can start writing software to control it. Any program which is to use the sound-box must contain a Rem statement in the first line. This statement will contain the machine-code routine. It is advisable, therefore, to add lines to the program. Alternatively, the lines can all be deleted, with the exception of the Rem statement.

As I mentioned, there are three controls to produce a sound, these are coarse, fine tune,

(continued on page 75)

(continued from page 73)

and length. For normal musical sounds it is advised to leave coarse tune set at 2. Poking fine tune with 0 will produce the highest note available, and with 255 will produce the lowest note. Poking length with 0 will deliver the longest note available and 255 will deliver the shortest note.

When the various controls have been set, the following line will actually make the noise:

LET SOUND = USR16514

To test the controls, use the same program as before and type Run 200. You will be invited to type in a number between 0 and 255 corresponding to the length of note required, and then invited to enter another corresponding to the frequency of the note required. It will then play the note.

```
HEX DUMP

OE OB 3E 00 16 00 14 BA C8
ED 49 21 FF 02 BC CA 88 40
2B C3 8F 40 S

BASIC PROGRAM

 1 REMXXXXXXXXXXXXXXXXXXXXXXXX
 2 LETX=0
 10 INPUTA$
 20 PRINTA$;
 30 IFA$="S"THENGOTO80
 40 LETB=16*CODEA$+CODEA$(2)-476
 50 POKE16514+X,B
 60 LETX=X+1
 70 GOTO10
 80 PRINT"TOP"
 90 STOP
 100 FAST
 110 POKE16519,INT(RND*255)
 120 POKE16526,INT(RND*255)
 130 LETSOUND=USR16514
 140 GOTO110
 200 FAST
 210 PRINT"LENGTH?"
 220 INPUTL
 230 POKE16519,L
 240 PRINT"NOTE?"
 250 INPUTL
 260 POKE16526,L
 270 LETSOUND=USR16514
 280 GOTO210
```

Maths teaching

M N Scarlett,
Maidenhead, Berkshire.

ZX-81

THE IDEA FOR this program originated from the Mathematics program in the Sinclair software catalogue, but it is a considerable improvement — particularly in the way the program seems friendly to the user. The user begins by entering his name which appears during the program run and the computer acknowledges by printing "Hello, I am the ZX-81" on the screen.

The function required, +—*/, is entered directly, rather than via a numeric code and then the user is asked by name for the level of difficulty he requires. Levels 1, 2 and 3 are offered on the screen but any numeric value can be inputted. 1, 2 and 3 produce sums capable of solution by mental arithmetic.

Lines 130 and 140 derive the two variables used in the question, "C" is related to "D" so that a negative result cannot be produced and in division the answer will always be a whole number. Line 150 assembles the sum into a string which is printed on the screen together with "=". This string consists of the variable "D" with the operator entered earlier, and the variable "C".

If the answer entered in response to printing the sum on the screen is correct, the score variable "S" is incremented and the user is told by name that his answer is correct and the score out of the number of sums attempted so far is printed. After a delay, the screen clears and a new sum is presented.

If the answer is wrong, "Wrong" appears on the screen and the sum is reprinted with the correct answer. For extra impact, "Wrong" can be programmed in reverse characters.

After 10 sums have been attempted, the user is told "Well done", and his score out of 10 is printed. This display remains on the screen for several seconds after which the program restarts automatically.

The Pause function has not been used within the main body of the program where delays were required since it tends to cause disturbances to the display. Instead, the single-line X = RND**RND**RND is used. A For-Next loop could have been used, but it would have required two lines for each delay. The program will run on a 1K ZX-81 and should provide a good deal of educational entertainment for the younger members of the family.

```
  5 REM "SUMS"
 10 PRINT "WHAT IS YOU NAME?"
 20 INPUT N$
 30 PRINT "HELLO ";N$;",",,"I AM THE ZX81"
 40 LET X=RND**RND**RND
 60 CLS
 70 PRINT "DO YOU WANT + - * OR / ?"
 80 INPUT A$
 90 PRINT "HOW DIFFICULT, ";N$;",","1,2 OR 3?"
100 INPUT B
105 LET S=0
110 FOR N=1 TO 10
120 CLS
130 LET C=1+INT(5**B*RND)
140 LET D=C*INT(5**B*RND)
150 LET B$=STR$ D +A$+STR$ C
160 PRINT B$;"=";
170 INPUT D
180 PRINT D
190 IF ABS (VAL B$ -D)>0.01 THEN GOTO 230
200 LET S=S+1
205 PRINT ,, "CORRECT,";N$;"."
206 IF N=10 THEN GOTO 280
210 PRINT S;" OUT OF ";N;" SO FAR."
220 GOTO 240
230 PRINT ,, "WRONG",B$; "="; VAL B$
240 LET X=RND**RND
260 NEXT N
290 PRINT "WELL DONE," ,, "SCORE";S;" OUT OF 10."
300 PAUSE 300
310 CLS
320 RUN
```

Zero dropper

G Stephen,
Aberdeen.

ATOM

THE FLOATING-point ROM in the Acorn Atom is very untidy when it comes to printing floating-point numbers on the VDU. The display takes the form 3.70000000 for the number 3.7 and the form 6.28000000E-12 for the number 6.28×10^{-n}. It would be much neater and more easily read if all the trailing zeros of the mantissa were dropped, and that is the job of this short subroutine.

The floating-point number to be printed is held in the variable ZN and the subroutine is called. The first line after the Rems — line 10010 — converts the number ZN to a string and stores it in the free RAM space starting at location 540 decimal.

If we assume that the number held in ZN is 6.28000000E—12, the string format will be:

```
6 = 2 8 0 0 0 0 0 0 E – 1 2 13
540                 550     554
```

There are two specific string areas used for this conversion; the main area is 540-554 decimal which holds the mantissa, and the secondary area 560-564 decimal holds the exponent.

Line 10010 also sets the secondary string to the null string by storing 13 at 560. The carriage return character 13 also signifies the end of a string. If there is an exponent, line 10030 is activated and proceeds to separate the mantissa from the exponent by moving the five bytes of data from 550-554 to 560-564. Then it stores 13 in 550 and so terminates the mantissa string.

```
6 = 2 8 0 0 0 0 0 0 E – 1 2 13
                    ↓        ↓
               560→ E – 1 2 13

6 = 2 8 0 0 0 0 0 0 13←550
```

We now have two strings $540 and $560 — $540 = 6.28000000 and $560 = E—12 — so we can proceed to remove those zeros in $540. Line 10050 does this by substituting the string terminator 13 for every zero until a non-zero value is detected.

Line 10060 then removes the decimal point if appropriate and the next line prints the two strings before returning to the main program. The value in ZN remains unchanged and can be used in further computations. This routine can be saved on tape and appended on to any program requiring a numerical printout. Here is a table of ASCII characters and their decimal values used:

ASCII	Decimal
carriage return	13
=	46
0	48
E	69

(continued on page 77)

(continued from page 75)

```
>10000 REM ROUTINE TO REMOVE TRAILING ZERO'S FROM A FLOATING-
>10005 REM POINT NUMBER(ZN). USES FREE RAM SPACE 540-564(DEC)
>10010 STR ZN,540 ; ?560=13
>10020 IF ?550<>69 THEN GOTO 10040
>10030 FOR I=550 TO 554 ; I?10=?I ; NEXT I ; ?550=13
>10040 I=549
>10050 DO ?I=13 ; I=I-1 ; UNTIL ?I<>48
>10060 IF ?I=46 THEN ?I=13
>10070 PRINT $540,$560 '
>10080 RETURN
>READY
```

Never a crossword

Iain Hancock,
Potters Bar,
Hertfordshire.

THIS PROGRAM, written for a ZX-81 — or a ZX-80 with the 8K Basic ROM — will print on the screen 20 anagrams of a word input by the user. The main section of the program, lines 130-180, removes a letter from the word, displays it and updates the original string.

It then repeats this until there are no more letters and returns to the beginning. The maximum length of a word on an unexpanded ZX-81 is about 10 letters, so a RAM pack would be useful.

```
5    REM *ANAGRAMS BY I.HANCOCK*
10   RAND
20   PRINT AT 6, 11; "ANAGRAMS"
30   SCROLL
40   PRINT"PHRASE?"
50   INPUT B$
60   CLS
70   PRINT TAB 10; "PHRASE-";B$
80   PRINT
90   FOR D=1 TO 20
100  LET A$=B$
110  LET A=LEN A$
120  FOR B=1 TO A
130  LET C=INT (RND*A)+1
140  PRINT A$ (C);
150  LET D$=A$ (1 TO C-1)
160  LET E$=A$(C+1 TO)
170  LET A$=D$+E$
180  LET A=A-1
190  NEXT B
200  PRINT
210  NEXT D
220  END
```

Half life

Raymond Nugent,
Aberdeen.

HALF LIFE may be of interest to science teachers. It was developed using the ZX-81 with 16K RAM. The program simulates the radioactive decay of a sample of 196 radioactive atoms which decay exponentially with respect to time.

The display shows the 196 atoms decaying randomly within a square and there is a continuous tally of the numbers of atoms which remain undecayed and those which have decayed as time passes. The display also gives an activity/time graph and the half life on two separate occasions.

Line 10 Sets up Dim space for 196 atoms. Variables X and Y give the number of decayed and undecayed atoms respectively. The variable E stands for the total number of attempts which the computer makes at finding an undecayed atom — this is equivalent to total time. "V" is used to give the correct number of "—" in subroutine 400.

"I" and "J" are selected randomly — if A$(I,J) is null then the display shows a pixel unplotted simulating the decay of that particular atom. A$(I,J) is then made equal to one and the tally altered. If A$(I,J) is not null, there is no change except in the time tally. A pixel showing the number of undecayed atoms is plotted every 14 attempts.

```
5    REM "HALF LIFE"
10   DIM A$(14,14)
15   LET E = 0
20   LET Y = 196
30   LET X = 0
40   LET V = 0
50   GOSUB 190
55   FOR N = 1 TO 14
60   LET E = E + 1
65   LET I = (INT(RND*14)+1)
70   LET J = (INT(RND*14)+1)
80   IF A$(I,J)=" " THEN GOSUB 500
90   PRINT AT 4,19;X;AT 6,19;Y; " "
91   PRINT AT 8,28; E
92   IF Y = 98 THEN GOSUB 400
93   IF Y = 49 THEN GOSUB 400
95   IF N = 14 THEN PLOT V+1,(INT(Y/6))+1
100  NEXT N
101  LET V = V + 1
110  GOTO 55
190  FAST
200  FOR B = 0 TO 63
210  PLOT B,0
215  PLOT 0,0.67*B
220  NEXT B
222  PRINT AT 4,8;" DECAYED:";AT 6,8; "UNDECAYED"
225  PRINT AT 20,28;"TIME"
228  PRINT AT 0,0;"ACTIVITY"
230  FOR R = 1 TO 14
232  PRINT AT 0, 22 + R/1.6;"▒" ;AT 8,22+R/1.6;"▒"
234  PRINT AT R/2,23;"▒" ; AT R/2,31; "▒"
240  FOR H = 1 TO 14
250  PLOT 47+R,27+H
255  NEXT H
260  NEXT R
265  SLOW
268  PAUSE 100
270  RETURN
400  FOR W= 1 TO (V/2)-1
410  IF Y = 98 THEN PRINT AT 12,W;"-";AT 20, (V/2)-1;E
420  NEXT W
430  RETURN
500  UNPLOT I+47,J+27
510  PLOT I+47,J+27
520  UNPLOT I+47,J+27
530  LET X=X + 1
540  LET Y=Y -1
550  LET A$(I,J) = "1"
560  RETURN
```

Palindromes

Tim Goldingham,
Maidenhead, Berkshire.

SEX AT NOON TAXES. Apart from its biological significance, this observation by John Julius Norwich is interesting because it is a palindrome — that is, it reads the same backwards as forwards.

Palindromes have always fascinated literary men. In a Charles Osbourne's recent biography of W H Auden, *The Life of a Poet,* the following is quoted:

T Eliot, top bard, notes putrid tang emanating, is sad, I'd assign it a name: gnat dirt upset on drab pot toilet.

This has 85 letters; but according to the Guinness Book of Records the longest palindrome, devised by Jeff Grant of New Zealand, has 11,125 words.

You may not be able to match that record, but you can have some fun with this ZX-81 program, which displays text both fowards and backwards on the screen as you type, thus:

SS
SEES
SEX XES
SEX AAXES
SEX AT TAXES
SEX AT NN TAXES
SEX AT NOON TAXES

Because the program uses Inkey$, the space key would be interpreted as Break, so the adjacent key is used for spacing: a full stop

(continued on next page)

(continued from previous page)

gives a space on the left, and a comma on the right: **, shifted H, clears the screen ready for you to try and create more new palindromes.

As an extra refinement, if the palindrome pivots about a single letter as in the Auden example, a single * will remove the duplicate. The Rubout key can be used in the normal way.

```
  1 REM PALINDROME GENERATOR
  2 COPYRIGHT T P GOLDINGHAM 1981
 10 CLS
 20 LET A$=""
 30 LET B$=""
 40 PAUSE 40000
 50 POKE 16437,255
 60 LET C$=INKEY$
 70 IF C$="**" THEN RUN
 80 IF C$="*" THEN GOTO 110
 90 IF C$<>CHR$ 119 THEN GOTO 130
100 LET A$=A$(TO LEN A$-1)
110 LET B$=B$(2 TO LEN B$)
120 GOTO 190
130 IF C$<>"." AND C$<>"," THEN GOTO 170
140 IF C$="." THEN LET A$=A$+"Δ"
150 IF C$="," THEN LET B$="Δ"+B$
160 GOTO 190
170 LET A$=A$+C$
180 LET B$=C$+B$
190 PRINT AT 10,0;"     32 spaces     "
200 PRINT AT 10,0;A$;B$
210 GOTO 40
```

Paying a mortgage

K Ward, Nottingham.

PET

MY WIFE and I had decided to move house, but we did not know how much this would cost in new mortgage repayments or how much we could loan from a building society. After enquiries, we found as a general rule that a first-time buyer could usually loan between two and two-and-a-half times his annual salary, and if you are on your second or third house you could loan only two to two-and-a-quarter times your annual salary, if loaned over about 20 years.

The next problem was to calculate what the monthly payments would be before and after tax relief. All this is done by our program which is written in Basic and on a 16K Pet 2001 series with the large keyboard. The whole program consumes 1,835 bytes of memory but there are many Rem statements. Also the line statements are kept short to make it easier to understand.

The reverse character "R" prints the statement in reverse characters. The reverse character "S" homes the cursor. The Tab () starts printing after the number of spaces indicated in the brackets — a maximum of 255 is allowed on the Pet.

INT(PNx100 + .5)/100

rounds up to the nearest two decimal places.

Line 200. A change in the For-Next loop in line 200 will alter the number of years displayed on the screen: 12 are shown on the Pet. Any more and information will scroll off the top of the screen.

```
5 REM-"T"=CURSER UP
10 REM-CHR$(147)=CLEAR SCRN.+CURSER HOME
15 REM-T=TAX-T2=TAX AS DECIMAL
20 REM-I=INTEREST-I1=MONTHLY INTEREST
25 REM-N=NO.OF YEARS--N1= NO.OF MONTHS
30 REM-PN=MONTHLY PAYMENTS
35 REM-X=SALARY
40 PRINTCHR$(147)
45 PRINTTAB(10) " ■MORTGAGE PROGRAM" PRINT
50 INPUT"WHAT IS YOUR ANNUAL SALARY ";X PRINT
55 INPUT" ARE YOU A FIRST TIME BUYER IF SO ANSWER'Y'OR'N' ";A$ PRINT
60 IFA$="N"THEN75
65 PRINT"AS A FIRST TIME BUYER YOU ARE USUALY     ALOUD 2.5 TIMES YOUR "
70 PRINTTAB(61)"T TANNUAL SALARY          WHICH IS ""■"2.5*X PRINT GOTO85
75 PRINT"AS YOU ARE NOT A FIRST TIME BUYER YOU    ARE USUALY ONLY ALOUD "
80 PRINTTAB(62)"TT 2.25 TIMES YOUR    ANNUAL SALARY WHICH IS ""■"2.25*X PRINT
85 INPUT "WHAT IS THE SIZE OF THE LOAN ";PV PRINT
90 INPUT"WHAT IS STANDARD RATE OF TAX ";T PRINT
95 INPUT"WHAT IS THE MORTGAGE INT. RATE";I PRINT
100 INPUT"HOW MANY YEARS IS THE LOAN FOR";N PRINT
105 T2= T/100 REM TAX AS DECIMAL
110 I1=(I/100)/12 N1=N*12 REM I1=MONTHLY INTAS DECIMAL-N1=NO.OF MONTHS FOR LOAN
115 P=PV*I1 REM INTEREST ON LOAN
120 PM=1-(1+I1)↑-N1
125 PRINTCHR$(147)
130 PN=P/PM REM-MONTHLY PAYMENTS
135 PRINT"■A LOAN OF£"PV"OVER"N"YEARS"
140 PRINT"THE MONTHLY PAYMENTS ARE"TAB(30)"■"INT(PN*100+.5)/100
150 SA=0
155 PRINT"■WITH TAX RELIEF MONTHLY PAYMENTS ARE "
160 FORJ=1TON
165 I2=PV*(I/100) REM INTEREST
170 T3=I2*T2       REM TAX RELIEF
175 S=T3+SA S=INT(S*100+.5)/100 REM TAX REL.
180 SA=S REM TOT SAVINGS
185 PS=PN-((PV*I/100)*T2)/12 REM PAYMENTS AFTER TAX REL.
190 PS=INT(PS*100+.5)/100
195 PV=PV+I2-(PN*12) REM LOAN AFTER ANN.PAYMENT
200 IFJ<13THENPRINT"■YEAR"J,"■"PS,"LOAN LEFT"INT(PV*100+.5)/100
205 NEXT J
210 PRINT PRINT"THE TOTAL AMOUNT PAID OVER"TAB(45)N"YEARS IS"
215 PRINTTAB(30)"■£"INT((N1*PN)*100+.5)/100
220 PRINT"T ASSUMING A TAX RELIEF OF"T"%                "THE AMOUNT PAID IS"
225 PM=PN*N1-S
230 PRINTTAB(30)"■£"INT(PM*100+.5)/100  END
```

Transposing music

J F Vincent, Reading, Berkshire.

ZX-81

THIS PROGRAM was developed to transpose music from one key to another. The string A$ contains the 12 notes of the chromatic scale — the flattened notes are represented by inverse characters.

The rest of the program converts the notes input in string B$ to integers or near-integers, then adds T, which is the number of semitones through which you want to transpose the music, and reads the appropriate note from A$.

Obviously, if you want to transpose down you either use (12-T) or change the plus sign in line 110 to a minus symbol. The program has been simplified in line 310 which will not return an integer — as you can see if you put line 130 to

PRINT P; A$(P)

Yet the number is treated as the next highest integer — this saves some space. Some interesting patterns can be generated by this program. For instance, put T=5, enter B$ as

"CDEFGABCDEFGAB"

then enter the result which the program prints out. Do this until your first B$ is repeated.

The program can also be used to compose music — albeit music of a rather modern nature — if T is replaced by a random function. For example, delete line 30 and change 110 to

LET P = P + INT (RND*10)

```
5      REM "TRANSPOSITION"
10     DIM A$ (12)
20     LET A$ = "AABBCDDEEFGG"
30     LET T = ?? (the number of semitones of transposition)
40     INPUT B$
50     FOR N = 1 TO LEN B$
60     LET P = CODE B$(N)
70     IF P>45 THEN GOTO 300
80     LET P = (P - 37)*2
90     IF P = 6 OR P = 8 OR P = 10 THEN LET P = P - 1
100    IF P> 10 THEN LET P = P - 2
110    LET P = P + T
120    IF P>12 THEN LET P = P - 12
130    PRINT A$(P);
140    NEXT N
150    PRINT
160    RUN
300    LET P = P - 165
310    IF P>= 2 THEN LET P = P + P/2
320    GOTO 110
```

Bomb disposal

Roger Brooks,
Binton Green,
Warwickshire.

ZX-81

THE ARTICLE on games applications for the ZX-80/81 in the August/September issue by Tim Hartnell prompted me to write this program which only just fits into 1K. The program maps the top left-hand corner of the screen with a 10-by-10 matrix. In it lies a point which corresponds to the location of a bomb. By entering various points in the normal co-ordinate fashion, values are printed which give an indication to where it is.

Obviously, a large number of values will lead to easy location and so the score is weighted accordingly and printed after the bomb has been diffused.

Those with 16K RAM packs can modify this program to include more elaborate Print statements and a combination problem to be solved before the bomb is diffused.

One tip I find useful when programming is the use of keywords in print statements. They save the valuable memory space in the ZX-81.

The words above the keys can be entered by the K cursor mode then editing, placing words if necessary in front. Those words in red can be entered by simply pressing shift, and it is this which must be used in line 19 — the word stop. In the list inverse characters are underlined.

```
1  FOR X=0 TO 9
2  PRINT AT 11,X;X
3  PRINT AT X,11;X
4  FOR Y=0 TO 9
5  PRINT AT X,Y;"_"
6  NEXT Y
7  NEXT X
8  LET A=INT(RND*9)
9  LET B=INT(RND*9)
10 LET G=40
11 PRINT AT 13,0;"DETONATION IN ";G;" CYCLES"
12 LET G=G-1
13 IF G=0 THEN GOTO 22
14 INPUT C
15 INPUT D
16 IF C=A AND D=B THEN GOTO 19
17 PRINT AT D,C;(INT((ABS(D-B)+ABS(C-A))/2)
18 GOTO 11
19 PRINT AT 17,0; "STOP DETONATION-PRESS D"
20 IF INKEY$="D" THEN GOTO 25
21 GOTO 20
22 CLS
23 PRINT "BOOM"
24 STOP
25 PRINT "BOMB SAFE - SCORE=";G**2
READY
```

Screen artist

T M Humphries,
Sutton Coldfield,
West Midlands.

ZX-81

PAGE 121 of the ZX-81 manual lists a program which draws a line from pixel (A,B) to pixel (C,D) but its 25 lines take nearly 1K of memory so it is little use as a subroutine for unexpanded machines.

My eight-line program does not only the same job, but leaves the initial values A, B, C and D unaltered and uses only four new variables. As this listing is intended as a subroutine you will first need to Input (A,B) and (C,D).

The program works on the principle that the distance between the "x" co-ordinates, X, and the "y' co-ordinates, Y, can be covered in M steps where M is the greater of the absolute values of X and Y. The distance to be travelled in each step is, therefore, X/M and Y/M respectively.

The appearance of very steep or very shallow lines can be improved by adding Step 2 to line 1040 and plotting every other point.

```
1000 LET X = C - A
1010 LET Y = D - B
1020 LET M = ABS X
1030 IF ABS Y > M THEN LET M = ABS Y
1040 FOR N = 0 TO M - 1
1050 PLOT A + N * X / M, B + N * Y / M
1060 NEXT N
1070 RETURN
```

Ark Royal

D Ewan,
Haddington,
East Lothian.

ZX-81

YOU ARE CHALLENGED to land an aeroplane on the aircraft carrier, *Ark Royal.* There is an obstruction at the start of the flight deck which you must not hit with the wheel of your aircraft — landing too hard on the flight deck also spells disaster.

Once mastered, the game can be made much harder by changing line 300. You now have to judge your height above the flight deck. If it is not correct, you will require to increase altitude to correct your descent rate. The program runs in 1K.

```
1Ø   LET S = Ø
2Ø   LET PY = INT (RND * 15) + 1
3Ø   FOR Q = 1 TO 2Ø
4Ø   FOR A = Ø TO 28 STEP 2
1ØØ  CLS
11Ø  PRINT AT 18, S + 4;  "1";  AT 19, S;  "▙] [ ]
     [ ] ▄▄ ██";  AT 2Ø, Ø;  " ----------------------------------"
12Ø  PRINT AT 2Ø, S; "▛▜ ▛▜ ▛▜ ▛▜ ▛▜ ▛▜ ▛▜"
13Ø  LET S = S + 1
14Ø  IF S = 27 THEN LET S = Ø
15Ø  PRINT AT PY, A;  "█▟ ▙▟ ▓▓ ▙▟*";  AT PY + 1, A + 2;  "O"
2ØØ  IF PY + 1 = 19 AND A + 3 = S THEN GOTO 420
21Ø  IF PY = 18 AND A = S THEN GOTO 4ØØ
22Ø  IF PY > = 19 THEN GOTO 42Ø
3ØØ  IF INKEY$ = "6" THEN LET PY = PY + 1
31Ø  IF INKEY$ = "7" THEN LET PY = PY - 1
32Ø  NEXT A
33Ø  NEXT Q
34Ø  STOP
4ØØ  PRINT AT Ø, 1;  "LANDED", Q; " ATTEMPTS"
41Ø  STOP
42Ø  PRINT AT Ø, 1;  "CRASHED"
43Ø  STOP
FOR A MORE ADVANCED GAME CHANGE LINE 3ØØ AS FOLLOWS:-
3ØØ  IF INKEY$ = "6" THEN LET PY = PY + 2
```

Christmas competition

BY ANTHONY ROBERTS

Round 1

1. When was the transistor invented?
2. The Millionaire was one of the first calculators sold commercially; when did it appear?
3. Babbage never succeeded in making his Analytical Engine work. IBM made the first machine to use all of Babbage's principles — when?
4. The first commerical computer was the Ferranti Mark 1 Star made at Manchester University. It consumed 27,000 kilowatts of power and had 4,000 valves. When was it announced?
5. The first digital calculator machine — it could add, subtract, multiply and divide — was invented in 1642. By whom?
6. One of the first electronic digital computers weighed in at 30 tons with 18,000 vacuum tubes and was called Eniac. What did the name stand for?

Round 2

What are the names of the Seven Dwarfs?

Round 3

What follows in these sequences?

1. 5 12 21 32 45 ...
2. 8 5 4 9 1 7 6 10 ...
3. F4E S9 ...
4. 1982 1988 1993 1999 2010 ...
5. 5 14 30 55 ...

Round 4

Flowchart Puzzle

YOU ARE asked to help sort out a micro evaluation kit which comprises four machines — Ant, Bat, Cat, Dog — each of which has a terminal and its very own connecting cable.

Naturally, everything is colour-coded. The wires are coloured ebony, fawn, gold and hazel; the screens are indigo, jade, kingfisher and lilac and the machines are mauve, navy blue, ochre and puce.

They have all been packed in a single, large box with computer-type instructions provided to enable one to set them up in a straight row with each screen sitting on its micro, ready to run. But the kingfisher screen is not mentioned. Can you say what colour machine it connects with? Here are the instructions provided.



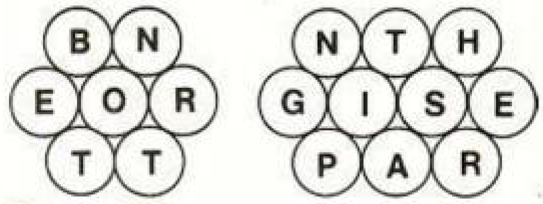
Round 5

Quotations

1. "Out of sight, out of mind" when translated into Russian by computer and back again became — what?
2. "I don't believe in maths" — who said it?
3. "Chips with everything" — who wrote it?
4. C P Scott said: "Television? No good will come of this device" Why?

Round 6

CAN YOU find the well-known phrases? Start at any letter and spell out the phrase by moving letter by letter into adjacent cells.



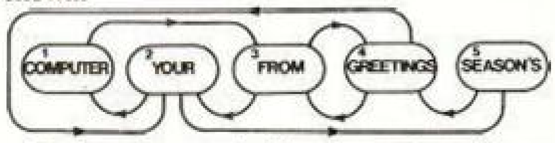
Round 7

WHAT PROCESS does this Basic program describe?

```
05 INPUT A,B
10 LET A=X
20 LET B=Y
30 LET Z=0
40 IF B<A THEN GOTO 80
50 LET A=A-B
60 LET Z=Z+1
70 GOTO 40
80 DISPLAY Z,A
90 STOP
```

Round 8

HERE ARE FIVE store locations which were intended to contain our Christmas message to you. Unfortunately, they have been put in reverse order. At least they can be moved along the connecting wires in the directions shown.



No location can hold two words, so if you move *Your* to *Computer* for instance, you must immediately move *Computer* to *From*. *From* can then move into the blank space left by your first move. Can you put the message right in just one sequence of 15 moves? Can you do it in less?

Round 9

IF YOU TAKE the digits of 1,634, take the fourth power of each and add them all together you get 1+1,296+81+256=1,634 — your original number. There are two other numbers which have this property — what are they?

Round 10

THE SOLUTIONS are all in front of your nose!

1. It goes rude.
2. Coe, your trump.
3. React to no ham.
4. Sam; free person.
5. Drop me, O comet!

A £15 book token will be awarded to the first correct solution. All entries must be at the *Your Computer* offices by the last working day in December. The result will be published in the February issue.

Competition reports

THANK YOU for sending in so many entries to the October ZX-81 Crossword competition. The winner is George Lang, Lonepine School Road, Romsey, Hampshire, who completed the sentence, "I will not use a ZX-81 to run a power station but be less 'ampered learning what's Watt in the computer age". A ZX-81 is on its way.

Almost everyone else followed a similar theme. A special mention though to J Duncan from Durham for writing, "but I did see the manager of Windscale in W H Smith".



Many readers complained that the first puzzle we ran in Competition Corner was too easy. In November we really got our own back with Crater Robots. Out of all the entries not one of you found the right solution — it was hard, though.

We have a solution, on the Sharp MZ-80K, which fits into about 2K of memory and took about nine minutes to run. It should be possible to write a solution to fit into a 1K Sinclair.

There are exactly 7,529,536,000,000 different ways for the robots to move, 101,566 of which produce results in which no two groups are in adjacent craters. If you tried to write a program that went through all these possible moves the run-time would take longer than was allowed for the competition. So a solution should concentrate on the different positions the groups could end up in. The correct number is just 168.

A mistake a lot of people make is to forget that there is no robot in the centre crater to start with. They then produce the wrong answer of 138. We do not have the space to publish the full program which gives the solution.

If you would like to see the program, write to us and we will post you a copy.

Congratulations to Jean Hartopp for trying hardest.



The ZX80/81 DESK CONSOLE

including switches for cassette and power
Complete and Assembled

- Fits your ZX80 or ZX81
- Just plug-in. NO soldering
- Switched for SAVE/RUN/LOAD
- Both cassette and computer inputs earthed when not functional
- Switch for your 9V supply
- LED monitor for supply
- Fits 16 RAM on 80 and 81
- Fitted with button cover
- Deep enough for mains plug cluster at rear (not supplied)
- Black grain finish in ABS plastic
- Recessed for manual/pencil

Price **£24.08** + **£2.00** p&p UK + VAT
Send your Name, Address, 80 or 81, together with cheque or PO only for **£29.99** to:

TTL BOX 2, WARMINSTER, WILTSHIRE BA12 7QX.
All orders acknowledged but allow 30 days for delivery.

Programs for your

ZX81

£3.95 each (post free)

from **A. Parsons, 23 Coxhill Gardens, River, Dover, Kent**

1K Learning Fun. Learn spellings, add in hours and minutes, price quantities of groceries or give change. *Plus* four function sums with whole figure answers and score. Plus Tables.

16K Learning Fun. Add in words, work out areas and perimeters, translate between French and English or enter words in any two languages. *Plus* vocabulary and two versions of Hangman.

16K Geography. Find towns in the U.K. or Europe with a series of five maps drawn on the screen. *Plus* Town and Country naming program and English Towns Hangman.

16K Art and Fun. Play Noughts and Crosses, Battleships or Board Game against the computer. *Plus* Draw a Picture. Print it if you have a printer, record it to display again, even add to it. *Plus* two fascinating pattern generators.

New titles continually being added

Fuller FD System for ZX80/81

The Fuller FD System is not a ZX80 81 'add on' but uses the micro board at the heart of a more powerfull system, allowing the user to expand with the system or stop with a standard keyboard & case and using existing Sinclair 16K RAM pack etc, either way you only need a screwdriver to assemble the built items, the FD Keyboard simply plugs into the ZX81 which is screwed in position in the FD case, ZX80 installation requires soldering to the keyboard, we will carry out this work at a fixed charge of £10. SAE will bring you details of our products. Have your ZX81 Kit built by us FREE when you buy The FD Keyboard, Case and Power Supply at £42.95 + £2.25 p & p

- FD Keyboard Kit £18.95
- FD Keyboard Built £24.95
- FD 16K RAM Board £39.95
- Above items + 80p p & p
- FD Keyboard/Motherboard add £15.95 to Keyboard price
- FD Case £11.75 + £1.25 p & p



To Fuller Micro Systems
Sandfield Park East
Liverpool L12 9HP

tick ZX80 ☐ 81 ☐

Please send me

I enclose a cheque/po for

Name

Address

the vic centre

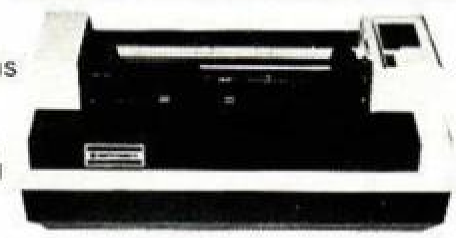
Adda Computers Ltd., a major supplier of computer systems to industry and business, have opened the Vic Centre in West London. Here you can see, discuss and buy everything to do with the new VIC 20 personal computer—in person or by mail. Hardware, software, technical advice and information is available from an experienced staff of experts. Even if you already own a VIC 20, get on our mailing list to know about new developments. Remember—everything has the backing of Adda's reputation, and there's a full 12-month warranty on all hardware. The Vic Centre is easy to reach—Just off the A40, close to North Acton tube station.

Not just a computer but a whole expandable system



AT ONLY £189·95 inc.VAT. Special cassette deck £44·95 inc.VAT.

The VIC 20 is a fully-fledged, easy-to-use computer. It's the core of a great expandable system, with full-size keyboard operation. First-time users can work it immediately with plug-in program cartridges, using your own colour TV to get up to 24 colours on screen, and three different sound tones. Or write your own programs in BASIC. The VIC 20 lets you build a system as needs and budget dictate. You can expand its memory to 32k Byte with Plug-in modules, and transfer data to external storage units. So the VIC 20 is more than just a personal computer—and its system will expand to put it even further ahead.



VIC 20 Dot Matrix Printer
Tractor-feed, 80 character-per-line, 30 characters-per-second printer.
£229.95 incl VAT.



VIC-RS232 INTERFACE
Fully implemented (true levels)
RS232C-V24
BI-DIRECTIONAL INTERFACE
Allows Vic to work as Mainframe Terminal
Drive a Qume Daisywheel or a Paper Tape Punch etc. etc.
FEATURE This unit contains master power supply which supports Vic's own supply when carrying Memory Expansions, Cassette Drives, Light Pens, Printers etc.
£56.35 incl VAT.



VIC-MEMORY 3K
Small size—low cost memory expansion.
Plugs into Vic and reproduces memory-port.
Can be used with other expansions gives a total of 6k user static ramon Vic.
FEATURE This board allows Vic to move Basic to begin at 1024 ($0400) as in Pet, and enables the use of HIGH RESOLUTION COLOUR GRAPHICS **£40.25** incl VAT.



VIC-TOOL KIT **£28.75** incl VAT.
For those who know tool kit on Pet we now have same facilities for Vic.
Renumber, Auto, Append etc.
This may be used with the stack VIC Rom Switch Board.

VIC ROM SWITCH BOARD £40.25 incl VAT.
An inexpensive unit which plugs direct on to Memory Expansion Port of the VIC and allows the insertion of up to 4 ROMS for games packs or toolkit aids, etc.
FEATURE. Simple software switch exchanges each pair of ROMS into VIC's ROM space allowing 'clashing' ROMS to be used.
FEATURE. Plug-in zero force sockets are available as optional extras to help eliminate pin damage to valuable ROMS.

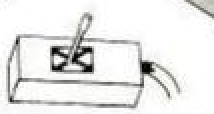
TERMS AND CONDITIONS: All goods sold subject to Adda terms and conditions of sale. Full details available on request, but include: 7 day money back guarantee. Adda 12 month hardware warranty. Please allow 21days for delivery. Allow 7 days for personal cheques to be cleared. Quoted prices are inclusive of VAT.



VIC-LIGHT PEN
This high quality light pen works in both normal and Hi-Res modes on the Vic allowing simple interaction with the Vic without keyboard entry.
Easy to program and easy to use.
e.g. Menu selection. Non-keyboard entry. Teaching Games.
FEATURE touch sensitive "Enter" contacts to eliminate accidental entry.
£28.75 incl VAT.



VIC-Games Port Adaptor Cable **£14.95** incl VAT.
games port plug.
A two into one adaptor for use with both joysticks and light pens. A must for those who require full control of games with graphics.
FEATURE low-cost High quality. Robust.



VIC JOYSTICK
Single
Hand-Held joystick units for games use available in Pair or Single configuration.
N.B. (2 Singles will not work as a pair unless modified)
£14.95 incl VAT.

Le Stick £30.75 incl VAT.
The ultimate joystick. One handed multi-directional super sensitive stick with built in fire button.

VIC software Each of these tapes **£14.95** incl VAT.
Codebreaker/Codemaker
You play the VIC or the VIC plays you in this computerised version of Mastermind.
VIC Seawolf, VIC Trap and Bounce-out
3 fun games, a submarine shoot out, a beat the VIC and an old favourite pub game. Good games with different skill levels.
Monster Maze and Maths Hurdler
A fun game with good colour and sound and a mental arithmetic learning game. Highly rated by everyone we have shown it to. Harder than you think.

Goods Required ____________ Price ______ YC/2

Add £2.00 p. & p. for orders under £50.00 Total £ ______

Name: ____________

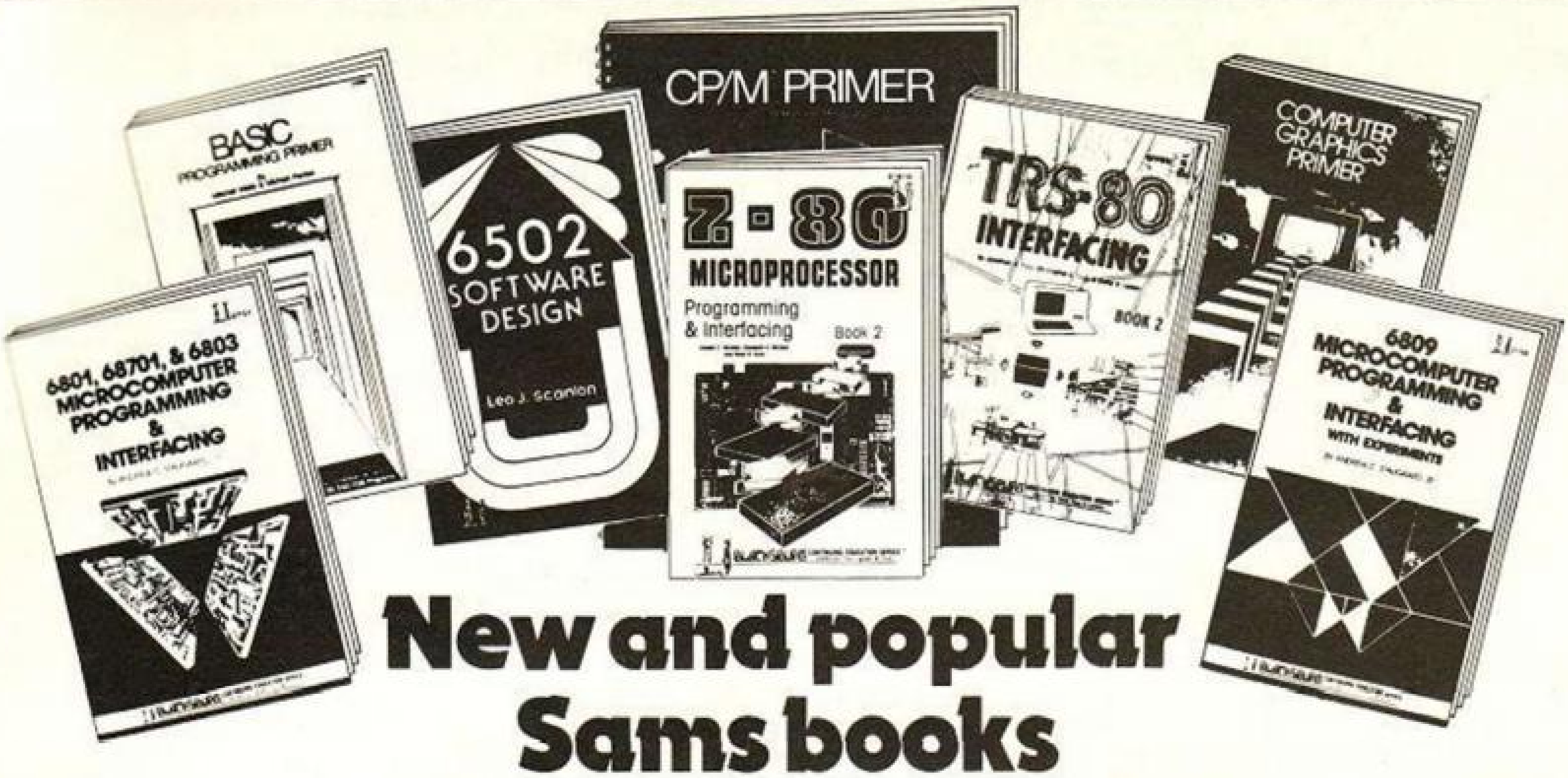
Address: ____________

SHOP ADDRESS: Adda Computers Limited, 154 Victoria Road, Acton, London, W3. Tel 01-992 9904.
OPEN: 10 am—6 pm (Tuesday—Friday), 10 am—5 pm (Saturday).

MAIL ORDER to: Adda Computers Limited, FREEPOST, London, W13 0BR or telephone your order (24 hours a day) to 01-992 9904 quoting your BARCLAYCARD OR ACCESS number.

* I enclose a cheque, made payable to Adda Computers Limited for
£ ....
* Please charge my Barclay/Access account. My account number is
....
* Please add my name to your mailing list
* Delete as applicable
Date: ....

adda



New and popular Sams books

Mitchell Waite and Michael Pardee

BASIC Primer

£8.35 672-21586-1

Mitchell Waite

Computer Graphics Primer

£10.45 672-21650-7

Stephen Murtha and Mitchell Waite

CP/M™ Primer

£10.45 672-21791-0

Leon Scanlon

6502 Software Design

£9.05 672-21656-6

Andrew C. Staugaard, Jr.

6801, 68701 and 6803 Microcomputer Programming and Interfacing

£10.45 672-21726-0

Andrew C. Staugaard, Jr.

6809 Microcomputer Programming and Interfacing with Experiments

£10.45 672-21798-8

Jonathan Titus, Christopher Titus and David Larsen

TRS-80 Interfacing: Book 1

£7.65 672-21633-7

TRS-80 Interfacing: Book 2

£8.35 672-21739-2

Elizabeth Nichols, Joseph Nichols and Peter Rony

Z-80 Microprocessor Programming and Interfacing: Book 1

£9.05 672-21609-4

Z-80 Microprocessor Programming and Interfacing: Book 2

£11.15 672-21610-8

Prices are correct at the time of going to press but may be subject to change. All titles advertised are published as paperback books.

Dealer enquiries are welcome:

Please contact Roy Jones at the address below or telephone Hemel Hempstead (0442) 58531

Prentice/Hall International

66 Wood Lane End, Hemel Hempstead,
Hertfordshire HP2 4RG, England.

Exclusive distributors of **Howard W. Sams books** in the UK and Europe.

Sams Books Stockists

Business & Electronic Machines
7 Castle Street
Edinburgh

Byteshop Computerland Ltd
P.O. Box 2
St Neots
Huntingdon
Cambridgeshire

Cambridge Computer Store
1 Emmanuel Street
Cambridge

Datron Micro Centre
Latham House
2 Abbeydale Road
Sheffield 7

Micro-C
5-11 Martineau Way
Union Street
Birmingham

Micro-C
Unit 2, Channons Hill
Industrial Estate
Fishponds
Bristol

Micro-C
57-59 Albion Street
Leeds

Micro-C
127 Charles Street
Leicester

Micro-C
Units 91-93 Arndale Centre
Luton
Bedfordshire

Micro-C
19 Brown Street
Manchester

Micro-C
31-35 Blagdon Road
New Malden
Surrey

Micro-C
2 Wheeler Gate
Nottingham

Micro-C
10-11 Bargate
Southampton
Hampshire

Newbear Computing Store
40 Bartholomew Street
Newbury Berkshire

Silicon Centre
Pictaural Electronics Ltd
21 Comely Bank Road
Edinburgh 4

Software House
Horseshoe Yard
Brooke Street
London W1

Tomorrow's World
Grafton Arcade
Grafton Street
Dublin 2

B.B.C. ROM Sets
Single Floppy Disk Units (100K)

ACORN COMPUTER

Write for details to: the Acorn Specialist in Yorkshire

New Hardware

- ACORN Atoms....................From £140.00
- ACORN GP-80 Printer + Cable....................£232.00
- ATOM Word Pack ROM....................£30.00
- Extra Memory per 1K (recommended)....................£3.20

New Software from ACORNSOFT

- GAME 1 Asteroids + Subhunt + Breakout....................£11.50
- GAME 2 Dogfight + Mastermind + Zombie....................£11.50
- GAME 3 Rat Trap + Lunar Lander + Black Box....................£11.50
- GAME 4 Star Trek + Four Row + Space Attack....................£11.50
- GAME 5 Invaders + Wumpus + Reversi....................£11.50
- GAME 6 Dodgems + Simon + Amoeba....................£11.50
- GAME 7 Green Things + Ballistics + Snake....................£11.50
- GAME 8 Stargate + Go + Moko + Robots....................£11.50
- SOFT VDU New character set design....................£11.50
- MATHS PACK 1. MATHS PACK 2....................£11.50
- UTILITY PACK 1. Disassemble + Fast COS + Renumber....................£11.50
- ATOM DATABASE. Versatile & efficient....................£11.50
- PEEKO-Processor Simulates micro....................£11.50
- ATOM FORTH. Full implementation....................£11.50
- ATOM FORTH. User's Guide....................£5.00

Books

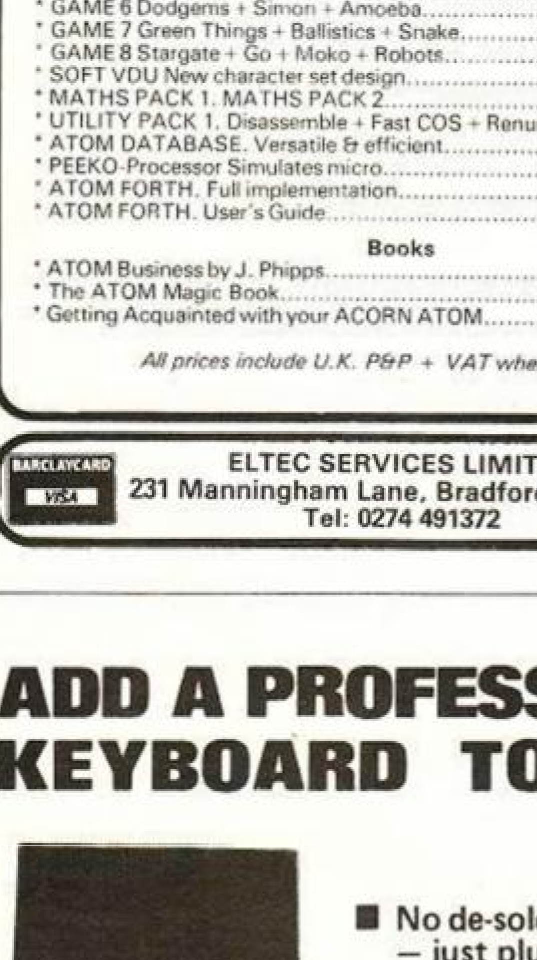
- ATOM Business by J. Phipps....................£6.95
- The ATOM Magic Book....................£5.50
- Getting Acquainted with your ACORN ATOM....................£7.95

All prices include U.K. P&P + VAT where applicable

BARCLAYCARD VISA

ELTEC SERVICES LIMITED
231 Manningham Lane, Bradford BD8 7HH
Tel: 0274 491372

ADD A PROFESSIONAL KEYBOARD TO YOUR ZX 81



- No de-soldering necessary — just plug-in.
- Full travel keyboard as used by international computer and terminal manufacturers.

£28·95
including VAT
post & packing

"All-you-need" easy to assemble kit comprises:

One piece 47 key keyboard module fitted but not soldered to double-sided printed circuit board; connectors; simple plug-in flexible cables; screws, mounting feet and legends. Two-part keycaps with buttons pre-assembled on keyboard and clear protective clip-on caps for fitting after legends have been positioned. Comprehensive assembly instructions supplied with each kit. Allow 28 days for delivery.

Please send ...... keyboard kit(s) @ £28.95 each, cheque or postal order made payable to:- YC Nov.
DK COMPUTER KEYBOARDS (dept.)
Glendale Park, Fernbank Road, Ascot, Berkshire.
Phone-a-Leaflet 03447 4731
Please ensure your name and address are clearly stated

Σ Sigma Technical Press

The UK Software Publisher

New Books:

BYTEING DEEPER INTO YOUR ZX81

by **Mark Harrison**

The ZX81 Microcomputer is now, debatedly, the fastest selling personal computer in the U.K. Unfortunately, the user's manual cannot answer all the questions and problems that arise when using the ZX81. Also, the user's manual gives the distinct impression of being written by an engineer, for engineers . . . not for the average user of the ZX81. The only way you can see the manual is by buying the machine!
BYTEING DEEPER INTO YOUR ZX81 supplements the ZX81 manual and provides an excellent introduction to computer programming. It starts from first principles, and, by reference to over 30 carefully graded examples, progresses to some of the most advanced techniques useable on this computer. It presents detailed projects and programs for the user to solve; each of these is suitable for domestic use (e.g. games, personal data banks, and homework aids).

0 905 10413 7 December 1981 150 pages £4.95

PRACTICAL PROGRAMS FOR THE B.B.C. COMPUTER AND THE ACORN ATOM

by **David Johnson-Davies**

Approved by Acorn for use with the ATOM, this book contains 20 practical programs for a wide range of different applications, ranging from mathematics and graphics, to language manipulation, and games. The programs are explained in great detail so they can be tailored to individual requirements, and many of them could be translated to run on other microcomputers.
The book is intended for owners of the Acorn ATOM and BBC Proton who understand how to enter and run programs, but do not necessarily consider themselves fully acquainted with BASIC or machine code. Many of the programs will run on minimum ATOMs, although some of the programs require a machine with the full 12k of memory.

0 905 10414 5 December 1981 125 pages £5.95

Please write for full details of the **SIGMA** *forthcoming publishing programme to: John Wilson, Product Manager,* **John Wiley & Sons Ltd.,** *Baffins Lane, Chichester, Sussex or to: Dr. Graham Beech,* **Sigma Technical Press,** *5, Alton Road, Wilmslow, Cheshire.*

Distributed by
John Wiley & Sons Limited
Baffins Lane · Chichester · Sussex PO19 1UD
England

Practical Computing

Britain's Leading Personal Computer Magazine

Practical Computing is read by businessmen, scientists, administrators, teachers, students. In fact by everyone who is interested in personal computers and wants to know more about them . . .

More about computer equipment. Down-to-earth reviews of personal computers and peripherals such as printers, terminals, VDU's and disc drives.

More about software. How to write it. What to look for in business software. Evaluations of software packages to take the guess-work out of your software purchases.

More about applications. What can you do with a personal computer? Case studies of the ways in which people as diverse as businessmen, manufacturers, doctors, scientists and teachers are using computers in their everyday work.

More about games. Designing your own games. How to achieve better graphics. Games and program listings to run on your own computer.

It all adds up to colossal value. The best and most informative selection of articles and regular features about personal computers for only 80p a month!

Practical Computing is available from leading newsagents.

Or complete the form and return it together with your cheque/postal order to obtain copies on subscription.

To: Marketing Services Dept., Rm 316, IPC Electrical Electronic Press Ltd., Quadrant House, The Quadrant, Sutton, Surrey SM2 5AS

Please send me a copy of Practical Computing every month for a year. I enclose a cheque/P.O. for £10 (UK)/£16 (Overseas) payable to IPC Business Press Ltd.

Name ....

Address ....

....

Acorn Atom

747



FLIGHT SIMULATION PROGRAM FOR THE 12K ATOM

Written for Bug-byte by a 747 pilot. Accurate simulation of a 747's cockpit display (airspeed, altitude, rate of climb, attitude, flaps, etc., and graphic display of horizontal situation & attitude); allows you to guide your craft to the landing strip. On making your final approach the display changes to a high-resolution 3D representation of the runway coming up to meet you. A real test of skill. Finding the runway is quite a challenge — landing safely is even more difficult. If you succeed, you are awarded a skill rating and the chance to take off and trying again.

REQUIRES FLOATING POINT ROM *PRICE ONLY £8.00*

VISA

ALL PRICES INCLUSIVE

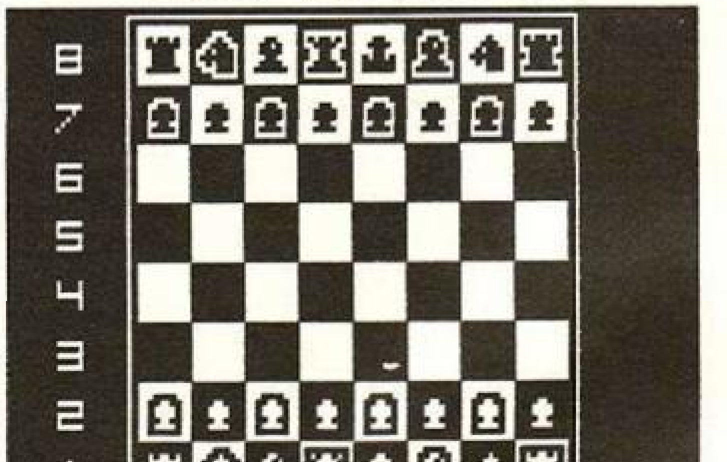
TELEPHONE 051 227 2642

BUG-BYTE

98-100 THE ALBANY OLD HALL STREET LIVERPOOL L3 9EP

Acorn Atom

CHESS



THE PROGRAM YOU'VE BEEN WAITING FOR!

Fantastic machine code chess game for the 12K Atom. Features include: split screen (high res. + alphanumerics); many levels of play; castling & en passant; computer plays black or white. Supplied on cassette with instructions. *PRICE ONLY £9.00*

DON'T FORGET — OUR PRICES INCLUDE VAT & POSTAGE

BUG-BYTE

98-100 THE ALBANY OLD HALL STREET LIVERPOOL L3 9EP

@@@@@@@ QUICKSILVA PRESENTS A RANGE OF QUALITY @@@@@@@
@@@@@ HARDWARE / SOFTWARE FOR THE @@@@@
@@@ @@@
@ @

ZX80/ZX81

HARDWARE -------- All hardware is sent with complete instructions and programming examples. Please allow 28 days for delivery.

QS MOTHER BD. -------------- ZX-80 / 81 ------------------- £10 : 00
Extends existing port to allow any Ram Pack + two other boards to be fitted. On board 5V. Regulator ; Two 23 way double sided edge connectors for add-ons.

QS 3K RAM BD. -------------- ZX 80 / 81 -------------------- £18 : 00
Reliable (pre-shrunk?) Static Ram Board which combines with the Computers internal 1K to give a total of 4K. Plugs direct into Computer or Mother Board.

QS SOUND BD. -------------- ZX 80 / 81 -------------------- £25 : 00
A programmable sound generator board using the versatile AY-3-8910.
3 Pitches / 3 Volumes / Noise source / two 8 bit Input ; Output Ports / Envelope shaper, all controlled from 'BASIC'. Plugs into any external amplifier.

QS CHARACTERS BD. ----------ZX 81 ONLY *---------------- £25 : 00
Simple Hardware mod (1 cut ; 1 resistor ; 1 wire) required to ZX-80.
Gives two programmable character sets of 64 characters each. Uses graphics key to shift between the two sets. Uses NO Ram space. Create your own characters - Upper and lower case alphabet ; scientific characters ; fine line graphics characters ; games characters (real space invaders). Works with existing programs and with the Printer. Price includes Demo Cassette.

QS CONNECTOR. ------------ZX-80 / 81 -------------------- £ 3 : 00
Consists of two 23 way double sided edge connectors back to back. One needed for any or all QS Expansions. (Except Ram Boards)

SOFTWARE ---------- All software is recorded twice on high quality cassettes and is sent complete with full operating instructions.

QS DEFENDER. ---- 3K RAM minimum ------- 4K or 8K ROM -----£ 5 : 50
Fast, Flicker free, Machine code, Moving graphics version of Arcade Game. Most complex moving graphics game yet for ZX-Computers. Up to 84 Fast moving characters on screen at once. First and only Full screen display.

QS LIFE. ----------4K RAM Minimum --------Zx-81 ONLY ----- £ 4 : 50
Fast program with Machine Code cell generation and screen display routine. Simulates the growth of living cells in a 20 * 32 Matrix. Random or programmed start positions plus M/C routine make this a fast, complex and varied program.

Send S. A. E. for FULL data sheets on all hardware and software. Cheques made payable to 'Quicksilva' and orders sent to the following address.------
QUICKSILVA : 95, UPPER BROWNHILL RD.: MAYBUSH : SOTON : HANTS.

THE

BUFFER

MICRO SHOP

(NEXT TO STREATHAM STATION)

★ ★ ★

NEW SOFTWARE SHOP EXCLUSIVELY FOR

ZX81

PROGRAMS, GAMES, "ADD-ONS"

★ ★ ★

MOST OF THE MAIL ORDER ITEMS ADVERTISED IN THIS MAGAZINE AVAILABLE OVER THE COUNTER

★ ★ ★

LOADING PROBLEMS? TRY OUR INTERFACE
BUSINESS & TECHNICAL DATA HANDLING PROGS.
PROPER KEYBOARDS; CONSOLES; VDUs

★ ★ ★

DUE TO OPEN

1st DECEMBER

BUT, PLEASE RING 01-274 6674 FOR CONFIRMATION AND FREE CATALOGUE

CAMBRIDGE LEARNING

SELF-INSTRUCTION COURSES

Microcomputers are coming - ride the wave! Learn to program. Millions of jobs are threatened but millions more will be created. Learn BASIC- the language of the small computer and the most easy-to-learn computer language in widespread use. Teach yourself with a course which takes you from complete ignorance step-by-step to real proficiency with a unique style of graded hints. In 60 straightforward lessons you will learn the five essentials of programming: problem definition, flowcharting, coding the program, debugging, clear documentation.



COMPUTER PROGRAMMING IN BASIC

£10.50

Book1 Computers and what they do well; READ, DATA, PRINT, powers, brackets, variable names; LET; errors; coding simple programs.
Book 2 High and low level languages; flowcharting; functions; REM and documentation; INPUT, IF....THEN, GO TO; limitations of computers, problemn definition.
Book 3 Compilers and interpreters; loops, FOR....NEXT, RESTORE; debugging; arrays; bubble sorting; TAB.
Book 4 Advanced BASIC; subroutines; string variables; files; complex programming; examples; glossary.

Understand Digital Electronics

Written for the student or enthusiast, this course is packed with information, diagrams and questions designed to lead you step-by-step through number systems and Boolean algebra to memories, counters and simple arithmetic circuits and finally to an understanding of the design and operation of calculators and computers.



DESIGN OF DIGITAL SYSTEMS £14.00

Book 1 Octal, hexadecimal and binary number systems; conversion between number systems; representation of negative numbers; complementary systems.
Book 2 OR and AND functions; logic gates; NOT, exclusive-OR, NAND, NOR and exclusive-NOR functions; multiple input gates; truth tables; De Morgans Laws; canonical forms; logic conventions; karnaugh mapping; three state and wired logic.
Book 3 Half adders and full adders; subtractors; serial and parallel adders; processors and ALU's; multiplication and division systems.
Book 4 Flip flops; shift registers; asynchronous and synchronous counters; ring, Johnson and exclusive--OR feedback counters, ROMS and RAMS.
Book 5 Structure of calculators; keyboard encoding; decoding display data; register systems; control unit; program ROM; address decoding.
Book 6 CPU; memory organisation; character representation; program storage; address modes; input/output systems; program interrupts; interrupt priorities; programming; assemblers; computers; executive programs; operating systems.

GUARANTEE No risk to you.
If you are not completely satisfied, your money will be refunded upon return of the books in good condition.

CAMBRIDGE LEARNING LIMITED, UNIT 89, RIVERMILL SITE,
FREEPOST, ST. IVES, HUNTINGDON, CAMBS., PE17 4BR, ENGLAND.
TELEPHONE: ST. IVES (0480) 67446

All prices include worldwide postage (airmail is extra - please ask for prepayment invoice). Please allow 28 days for delivery in U.K.

Please send me the following books:-

.... Computer Programming in BASIC @ £10.50

.... Design of Digital Systems @ £14.00

I enclose a *cheque/PO payable to Cambridge Learning Ltd
for £.................... (*delete where applicable)
Please charge my:

*Access/American Express/Barclaycard/Diners Club/Eurocard/Visa/Mastercharge/Trustcard

Expiry Date.......... Credit Card No...........................

Signature..........................
Telephone orders from credit card holders accepted on 0480 67446 (Ansafone).
Overseas customers (incl. Eire) should send a bank draft in sterling drawn on a London bank, or quote credit card number.

Name..

Address...

....

Cambridge Learning Limited, Unit 89, Rivermill Site,
FREEPOST, St. Ives, Huntingdon, Cambs. PE17 4BR, England.
(Registered in England, No. 1328762)

The Exhibition Which Works For You

MICROSYSTEMS '82

WEST CENTRE HOTEL, LONDON FEBRUARY 24-26, 1982

Over 6300 quality visitors attended the 1981 show — providing the correct balance of users and specifiers of your products and services. The formula is right — you can make MICROSYSTEMS '82 work for you by reserving your stand space NOW.

Find out how exhibiting at MICROSYSTEMS '82 can work for you by completing and returning the coupon now, to:
Exhibition Manager, MICROSYSTEMS '82, IPC Exhibitions Ltd., Surrey House, 1 Throwley Way, Sutton, Surrey SM1 4QQ.

MICROSYSTEMS '82 is sponsored by Computer Weekly, Systems International, Practical Computing, Your Computer, Computer Talk, Office Systems, Data Processing and Microprocessors and Microsystems and organised by IPC Exhibitions Ltd.

Please send details of exhibiting at MICROSYSTEMS '82, to:

Name ______________________

Position in company ______________________

Company ______________________

Address ______________________

______________________ Tel. No. ______________

ZX81 SOFTWARE (16K)

TITLE & DESCRIPTION	*CODE*	*PRICE*
ZX81 BUSINESS SOFTWARE		
VIDEO-AD. Continuously rotating display of 16 pages of advertising or	AD81S (Std)	7.95
other information. Shop window/ classroom/exhibition/seminar.	AD81L (Lux)	9.95
VIDEO-PLAN. Financial modelling package. Scaled down to fit the	PLAN81S (Std)	7.95*
ZX81 computer. Space for up to 1000 × 10 digit numbers plus headings and titles.	PLAN 81L (Lux)	9.95*
ZX81 SERIOUS SOFTWARE		
VIDEO-GRAPH. Use the computer to build pictures and merge them	GRAPH81S (Std)	5.95
like an identikit.	GRAPH81L (Lux)	7.95
VIDEO-VIEW. Your own personal and private version of teletext/	VIEW81S (Std)	5.95
viewdata.	VIEW81L (Lux)	7.95
VIDEO-MAP. A game with a serious purpose. A geographical tutor	MAP81S (Std)	5.95*
based on maps.	MAP81L (Lux)	7.95*
ZX81 GAMES PACKS		
FORCE-FIELD. Hostile UFOs attack a city. You control the force field which destroys their bombs.	FORCE	3.95*
SPACE-RACE. Party game for up to eight players.	SPACE	3.95*
FOOTBALL-LEAGUE. For the student of football. Simulate an entire season's play.	FOOT	3.95
TEST-MATCH. For the student of cricket. Select your teams and see who wins.	TEST	3.95
STOCK-MARKET. Buy and sell your way to a fortune.	STOCK	3.95*

*Not suitable for ZX80 with 8K ROM.

Send large s.a.e. for further details.

To: **Video Software Ltd.**
Stone Lane, Kinver,
Stourbridge, W. Midlands.
DY7 6EQ

ORDER FORM

YC12

CODE	TITLE	QTY	PRICE	TOTAL

Cheque/PO No. .... For

NAME..

ADDRESS..

....

....

CASSETTE ONE

PROGRAMS FOR ZX81

"I had your Invaders/React cassette . . . I was delighted with this first cassette."

P. Rubython, London

"Thanks for your Cassette One you sent me — some excellent games at a very cheap price!"

P. Rushton, Leeds

MACHINE CODE	**BASIC**
REACT	1 CHING
INVADERS	MASTERMIND
PHANTOM ALIENS	BASIC HANGMAN
MAZE OF DEATH	ROBOTS
PLANET LANDER	
BOUNCING LETTERS	
BUG SPLAT	

Ideal if you have a 1K ZX81 because:

- All the ACTION programmes are in machine code
- No-one else sells machine code programs as cheaply
- Quality cassettes (FUJI or TDK) are used
- The sequence of 11 programs is repeated 4 times, just in case a program won't load
- Cassette One is posted First Class
- Side 2 contains large screen versions of INVADERS and MAZE OF DEATH, ready for when you get the 16k RAM pack

Send £3.80 to

Michael Orwin, 26 Brownlow Road, Willesden, London NW10 9QL

P.S. Previous customers who did not get the large screen versions (for 16K) can get free upgrade instructions by sending me a sae.

MICROCOMPUTER COMPONENTS

LOWEST PRICES – FASTEST DELIVERY

Device	Price
MEMORIES	
2114L-200ns	1+1.28
	25+1.19
2114L-300ns	1+1.28
	25+1.19
2708 450ns	1.99
2716 450ns	1+2.49
	25+2.37
2716 350ns	6.95
2532 450ns	1+5.50
	25+5.31
2732 450ns	1+4.80
	25+4.08
4116 150ns	1+1.15
	25+1.06
4116 200ns	1+0.80
	25+0.72
6116 200ns	10.95
4118 200ns	3.90
6264 200ns	12.00
5516 200ns	22.88
CRT CONTROLLERS AND GRAPHIC DISPLAY PROCESSORS	
EF9364	5.94
EF9365	62.90
EF9366	62.90
BUFFERS	
81LS95	0.90
81LS96	0.90
81LS97	0.90
81LS98	0.90
8T26A	1.50
8T28A	1.50
8T95	1.50

Device	Price
8T97A	1.50
8T98	1.50
DATA CONVERTERS	
ZN425E-8	3.50
ZN426E-8	3.00
ZN427E-8	6.28
ZN428E-8	4.78
ZN429E-8	2.10
ZN432CJ-10	28.09
ZN433CJ-10	22.59
ZN440	56.63
MISCELLANEOUS	
AY-3-1015	3.90
AY-5-1013	3.45
AY-5-2376	6.95
MC1488	0.64
MC1489	0.64
MC14411	6.94
MC14412	7.99
RO-3-2513L	7.70
RO-3-2513U	7.70
ZN450E	7.61
7805	0.50
7812	0.50
7905	0.55
7912	0.55

Device	Price
ZILOG Z80 FAMILY	
Z80 CPU	4.00
Z80A CPU	4.82
Z80 CTC	4.00
Z80A CTC	4.00
Z80 DART	7.18
Z80A DART	7.18
Z80 DMA	11.52
Z80A DMA	9.99
Z80 PIO	3.78
Z80A PIO	3.78
Z80 SIO-0	13.95
Z80A SIO-0	13.95
Z80 SIO-1	13.95
Z80A SIO-1	13.95
Z80 SIO-2	13.95
Z80A SIO-2	13.95
EFCIS 6800 FAMILY	
EF6800	3.70
EF6802	5.11
EF6803	11.80
EF6809	11.95
EF6810	1.35
EF6821	1.74
EF6840	4.20
EF6845	9.50
EF6850	1.70

Device	Price
EF6862	6.91
EF6871-A1T	18.70
EF6880	1.07
EF6887	0.80
EF68488	9.11
EF6875	4.18
6502 FAMILY	
SYP6502	4.95
SYP6520	3.15
SYP6522	4.95
SYP6532	7.95
8080 FAMILY	
8085A	5.50
8212	1.70
8216	1.70
8224	2.45
8228	3.95
8251	3.95
8253	7.95
8255	3.95
LOW PROFILE DIL SOCKETS	
Number of Pins:	
8	0.07
14	0.09
16	0.09
18	0.15
20	0.17
22	0.21
24	0.23
28	0.25
40	0.29
CRYSTALS	
1 Mhz	3.00
1.8432 Mhz	2.50
4 Mhz	1.65

CHRISTMAS SPECIAL OFFER

10% DISCOUNT ON ANY ORDER IF ACCOMPANIED BY THIS ADVERTISEMENT

OFFICIAL ORDERS WELCOME — CREDIT CARD ORDERS WELCOME — **QUANTITY DISCOUNTS AVAILABLE**

All prices exclude post and packing on orders under £10 (50p) and VAT (15%).
ALL ORDERS DESPATCHED ON DAY OF RECEIPT WITH FULL REFUND FOR OUT OF STOCK ITEMS IF REQUESTED.
24-hour Telephone Credit Card Orders

MIDWICH COMPUTER CO. LTD.

(Dept YC)

HEWITT HOUSE, NORTHGATE STREET, BURY ST. EDMUNDS, SUFFOLK IP33 1HQ
TELEPHONE: (0284) 701321 TELEX: 817670

CHROMASONIC electronics

DEPT Y.C., 48 JUNCTION ROAD, ARCHWAY, LONDON N19 5RD

100 yds FROM ARCHWAY STATION & 9 BUS ROUTES

TELEPHONE 01-263 9493 263 9495

YOUR SOUNDEST CONNECTION IN THE WORLD OF COMPUTERS

PET



phone for prices

4016 16K RAM
4032 32K RAM
4040 Dual Drive Disk

The new PET printer.
4022 80 column tracks feed.
3023 80 column friction feed.
C2N Cassette Unit.

For the business man we stock the **8000** range inc. **8032** and **8050** with daisy wheel printers coming soon.

32K SYSTEMS AVAILABLE FROM ONLY £1,499

VIDEO GENIE



Utilises Z80, 12K level II Basic, Integral Cassette Deck, UHF O/P, 16K RAM, all TRS80 features. Simply plugs into monitor or UHF TV. With V.U. Meter. NOW WITH LOWER CASE AS STANDARD **£279**

PARALLEL PRINTER INTERFACE INC. CABLE	£33.00
CHROMASONICS PROGRAMABLE SOUND KIT	£24.94
SOUND KIT (FITTING EXTRA)	£7.00
LOWER CASE KIT (FITTING EXTRA)	£27.50
COLOUR KIT (FITTING EXTRA)	£34.95
EXPANSION BOX WITH/WITHOUT RS232	£215/185
16K/32K RAM CARD	£94/129
NEW GENIE II NOW AVAILABLE	£299.00

APPLE

APPLE II PLUS

Apple II plus apple

48K Machines **£595**
Disk Drive with Controller **£349**
Disc Drive without Controller **£289**
Colour Card **£69**
Graphics Tablet **£425**
Silent type printer **£199**
48K system available from **£1,786**



PRINTERS



EPSON MX80 £359
Dot-matrix printer with Pet graphics interface. Centronics parallel and serial. Pet and Apple compatible. True bidirectional, 80 cps.

EPSON MX80 FT/1 £399
Dual single sheet friction and tractor, 9 wire head, true descenders.

INTERFACES AND CABLES
for Apple II, Pet, TRS80, RS232, UK101, Sharp Superboard all available.

EPSON MX80 FT/2 £440
An FT/1 with high resolution graphics.

EPSON MX100 £570
Friction and tractor feed, high resolution graphics, wide carriage 15½ inches.

SEIKOSHA GP80A £199
Dot matrix 5 × 7, 80 columns 30 cps. graphics, double width characters.

MONITORS

GREEN MONITOR 9"	£98.00
12" BMC Green	£175
Hitachi professional monitors	
9" Black & White	£99.95
12" Black & White	£149.00



UK101

DOWN IN PRICE

UK 101 Kit **Inc 8K memory**	£125
Ready Built **Inc 8K memory**	£175
Complete in case	£199
4K Expansion 8 × 2114	£10
Memory Expansion Kit	
Printer Interface	£24.50
Sound generator plus	
PIO kit	£29.95
Cases	£19.95

NEW * NEW * NEW

Chromasonic Sound Kit	£24.50
Colour Kit	£69.95
32K Dynamic Memory Expansion Kit	£89.95
Inc. Demo Tape & Full Documentation. Send for details	
PIO and Eprom programmer kit	£24.50

VIC 20

VIC 20

Colours
24 total. 8 for characters, 8 for border, 16 for screen mixed as you wish. Basic colours on program keys are black, white, red, blue, light blue, green, yellow, and purple.

Sound
3 Tone Generator for music
"White Noise" Generator for language and sound effects.
Each Generator gives 3 octaves.
Reproduction is through TV speaker.

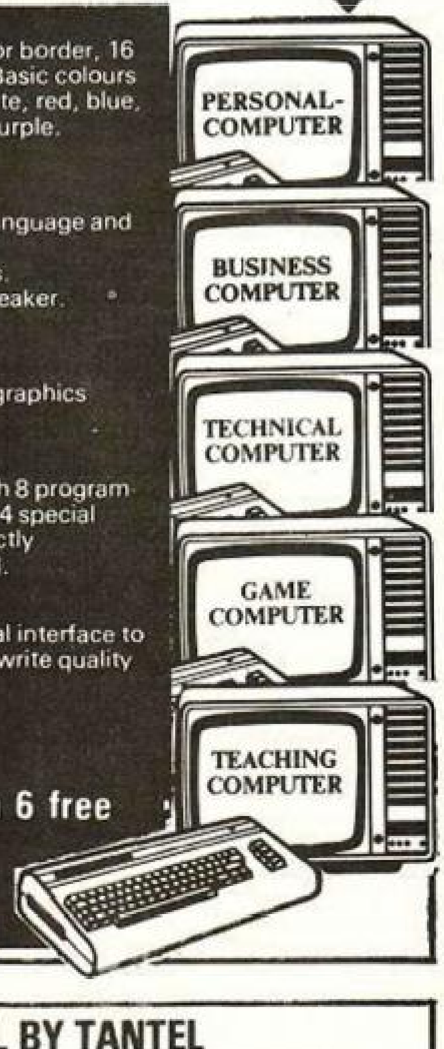
Character/ Line Display
22 Characters by 23 lines
64 ASCII characters, pet-type graphics character set.

Keyboard
DIN typewriter keyboard with 8 programmable function possibilities via 4 special function keys. Colours are directly addressable from the keyboard.

Peripherals/ Accessories
VIC Datacassette with special interface to guarantee high reliability read/write quality (PET/CBM compatible).

PRICE ONLY £165
CASSETTE DECK with 6 free programmes ONLY £34.75

Send for list of VIC-20 peripherals



TANTEL

PRESTEL BY TANTEL

COMMUNICATION AT YOUR FINGER TIPS FOR BUSINESS & HOME. UP TO DATE INFO

180,000 pages of information on Travel, News, Investment, Holidays, Hotels Etc., Etc.

only £159

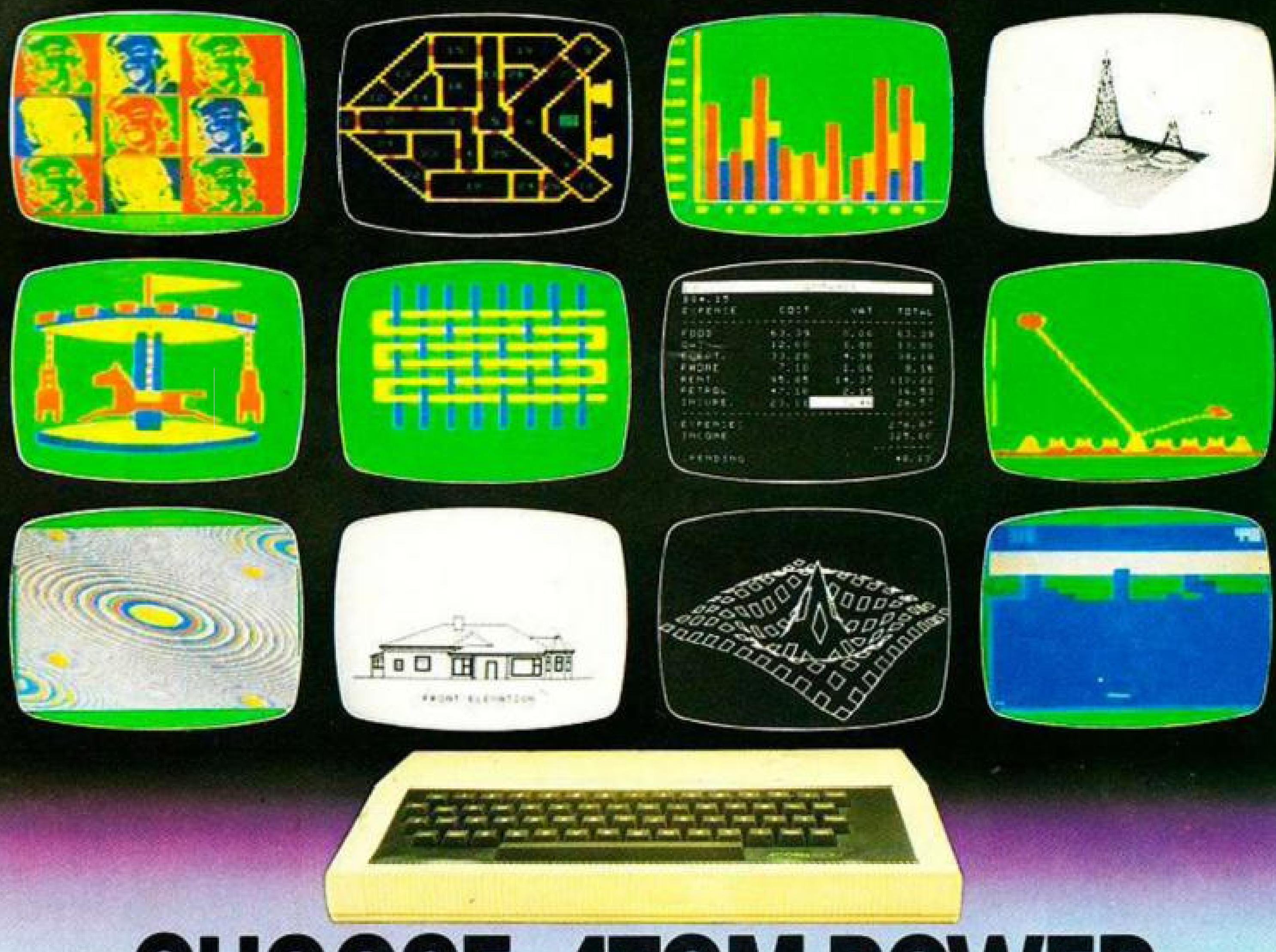
TANTEL IS POST OFFICE APPROVED. SEND FOR DETAILS.
DEMONSTRATION AVAILABLE AT OUR SHOWROOM

ALL ITEMS CARRY A ONE YEAR GUARANTEE

BARCLAYCARD VISA

Please add VAT 15% to all prices. Postage on computers, printers and cassette decks charged at cost, all other items P&P 30p. Place your order using your Access or Barclaycard. (Min. tel order £5). Trade and export enquiries welcome.

ADVERTISEMENT INDEX



CHOOSE ATOM POWER

At work or play – everything you need in a personal computer

The Atom is a machine to be used. Every day, day after day. It's a full function machine – check the specification against others. It's rugged, easy to operate built to last and features a full-size typewriter keyboard.

Just look at some of the features!

- More hardware support than any other microcomputer ● Superfast BASIC – can be updated to BBC BASIC if required
- High resolution and comprehensive graphics ideal for games programmers and players* ● Integral printer connection*
- Software available for games, education, maths, graphs, business, word processing, etc.
- Other languages: Pascal, FORTH, LISP
- I/O port for control of external devices
- Built-in loudspeaker ● Cassette interface
- Full service/repair facility ● Users club

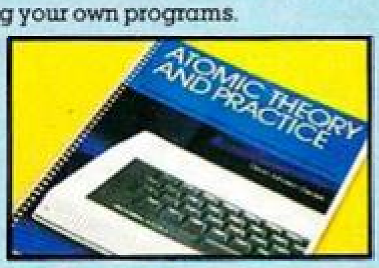
* Expanded version only

Optional Extras

- Network facility with Econet
- Disk ● PAL UHF colour encoder
- Add-on cards include 32K memory, analogue to digital, viewdata VDU, disk controller, daisywheel printer, plus many, many more! ● Power supply

FREE MANUAL

The Atom's highly acclaimed manual comes free with every Atom and leaves nothing out. In just a while you'll be completely at ease with your new machine! Within hours you'll be writing your own programs.



YOU AND YOUR CHILDREN

More and more schools are buying Atoms. More and more children will learn on an Atom. You can give them that extra familiarity with an Atom in the home.

ACORN COMPUTER 4a Market Hill, CAMBRIDGE CB2 3NJ

When you order your Atom we will include full details of all software packs and the optional hardware.

To: Acorn Computer Limited, 4A Market Hill, Cambridge CB2 3NJ.

I enclose a cheque/postal order for £........

Please debit my Access/

Barclaycard No ....

Signature ....

Name (please print)........................

Address ....

....

Telephone Number ....

Registered No. 1403810 VAT No. 215 400 220

Quantity	Item	Item price inc. VAT + P&P	Totals
	Atom Kit 8K ROM + 2K RAM	@ £140.00	
	Atom Assembled 8K ROM + 2K RAM	@ £174.50	
	Atom Assembled 12K ROM + 12K RAM	@ £289.50	
	Power Supply	@ £ 10.20	
		TOTAL	

ATOM SOFTWARE is designed and produced by Acornsoft, a division of Acorn Computers, Trust the manufacturer to get the very best from its own product. Current software includes word processing, maths packs over 30 games, database, Forth and business packages.



Write to Acornsoft, 4a Market Hill, Cambridge for full details and prices.

SEE OUR DEALER LIST ON PAGE 98